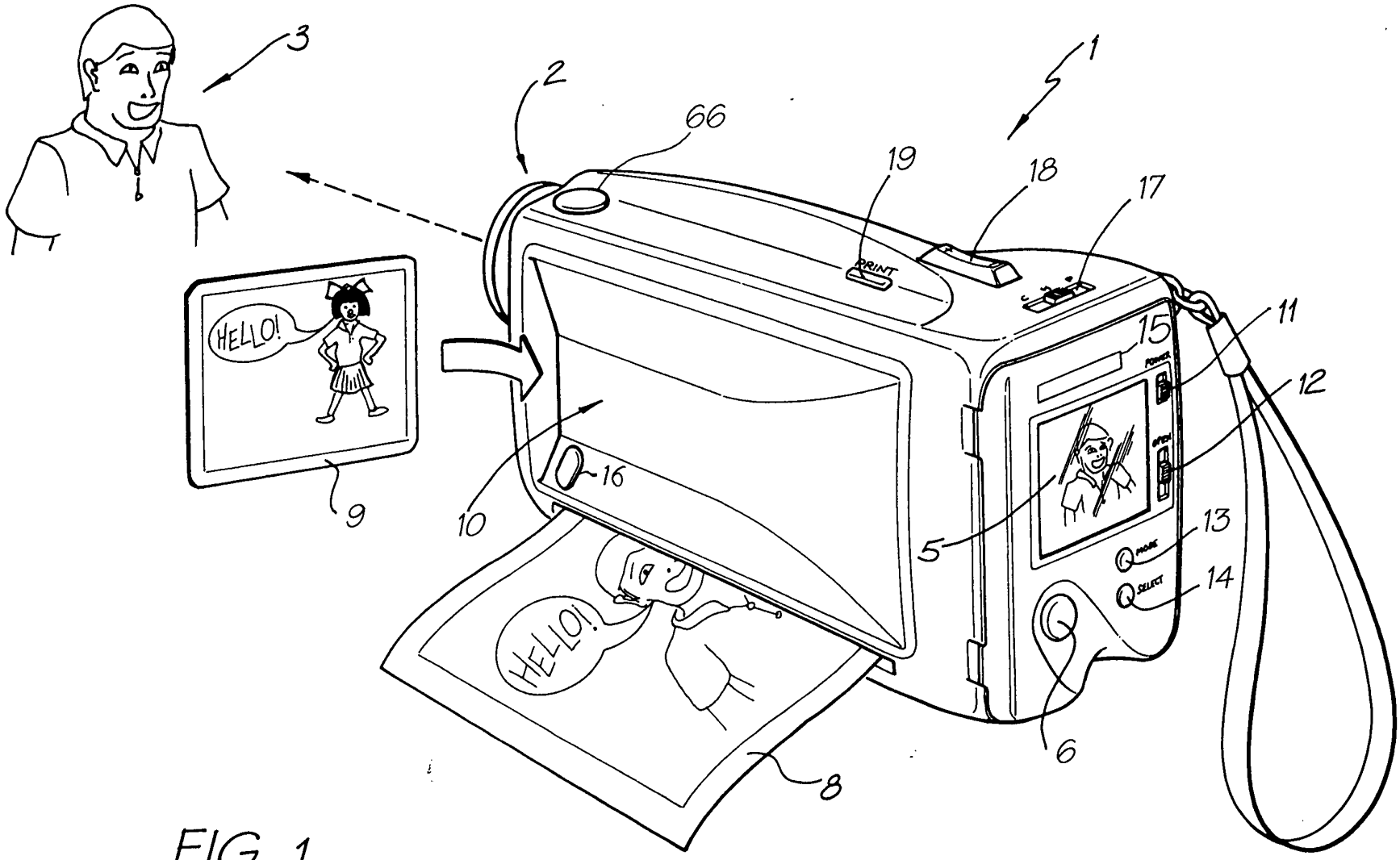


FIG. 1

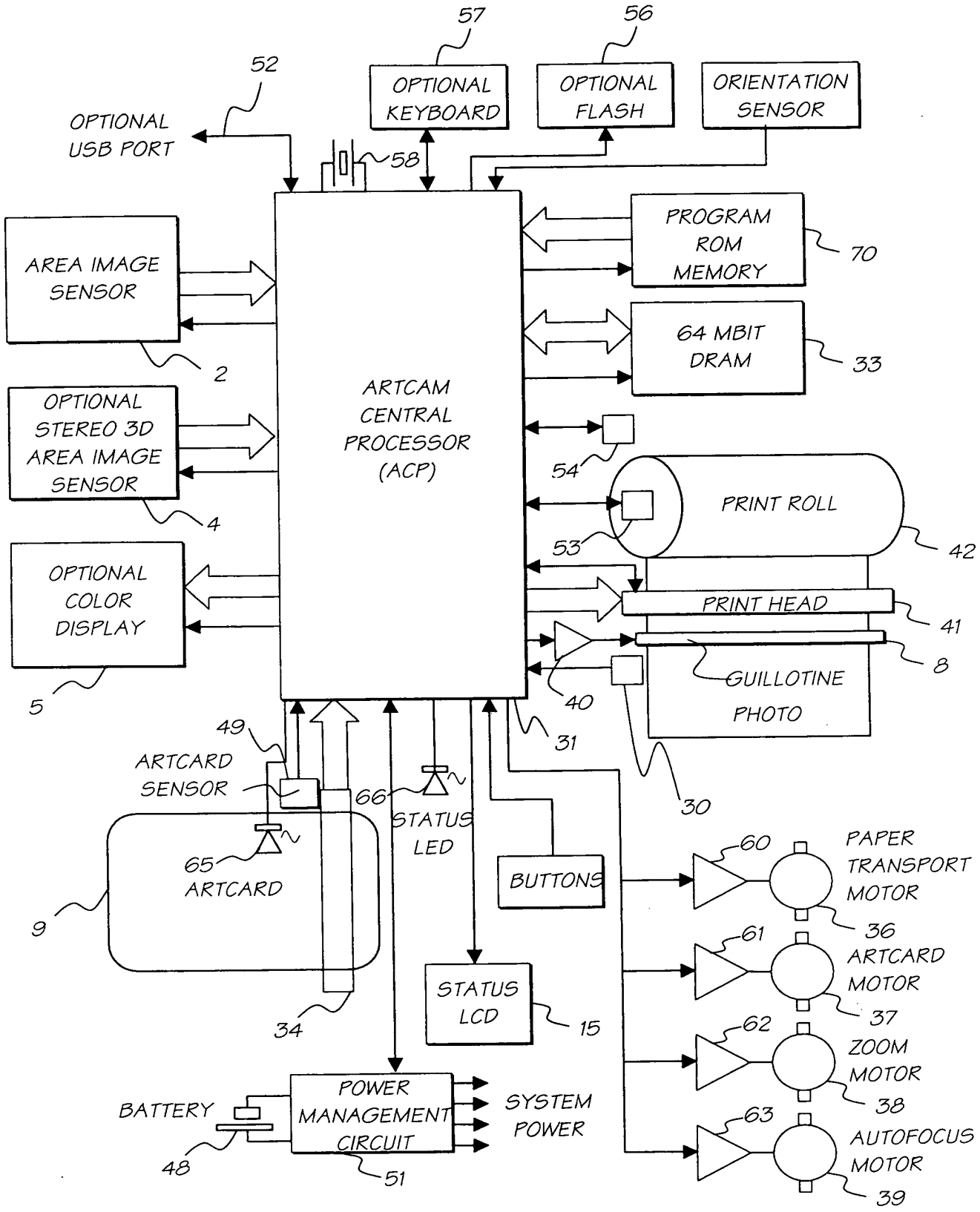

57
56
52
OPTIONAL KEYBOARD
OPTIONAL FLASH
ORIENTATION SENSOR
OPTIONAL USB PORT
58
PROGRAM ROM MEMORY
70
AREA IMAGE SENSOR
64 MBIT DRAM
33
2
ARTCAM CENTRAL PROCESSOR (ACP)
OPTIONAL STEREO 3D AREA IMAGE SENSOR
54
PRINT ROLL
4
53
42
OPTIONAL COLOR DISPLAY
PRINT HEAD
41
8
GUILLOTINE
PHOTO
5
40
49
31
ARTCARD SENSOR
66
STATUS LED
30
65
ARTCARD
BUTTONS
60
PAPER TRANSPORT MOTOR
9
36
61
ARTCARD MOTOR
37
STATUS LCD
15
34
62
ZOOM MOTOR
38
POWER MANAGEMENT CIRCUIT
BATTERY
SYSTEM POWER
63
AUTOFOCUS MOTOR
48
51
39

FIG. 2

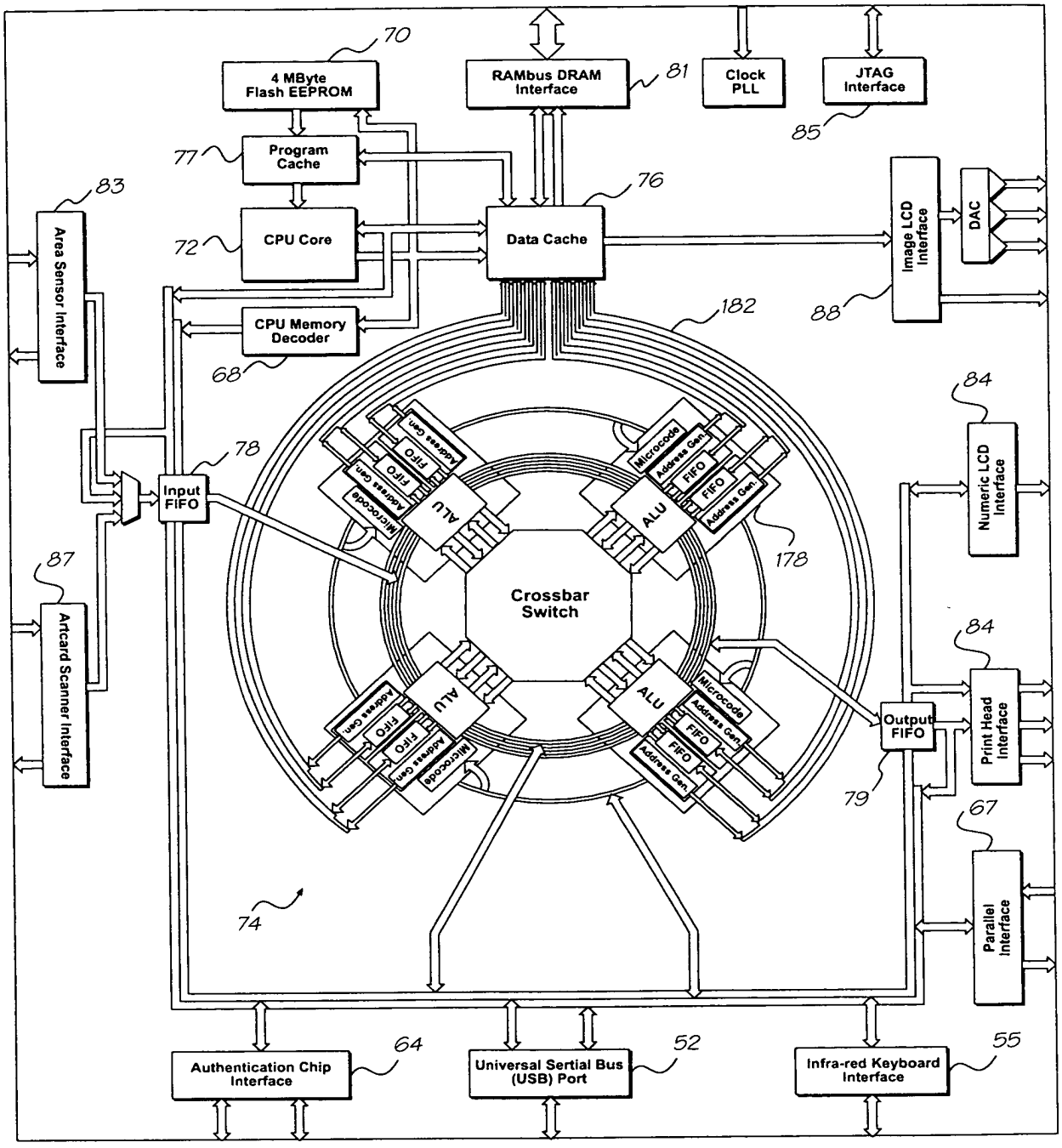

70
4 MByte Flash EEPROM
RAMbus DRAM Interface
81
Clock PLL
JTAG Interface
85
77
Program Cache
83
Area Sensor Interface
72
CPU Core
76
Data Cache
Image LCD Interface
DAC
88
182
CPU Memory Decoder
68
84
Numeric LCD Interface
78
Input FIFO
87
Artcard Scanner Interface
Crossbar Switch
ALU
Microcode
Address Gen.
FIFO
178
84
Print Head Interface
Output FIFO
79
67
Parallel Interface
74
Authentication Chip Interface
64
Universal Sertial Bus (USB) Port
52
Infra-red Keyboard Interface
55
31

FIG. 3

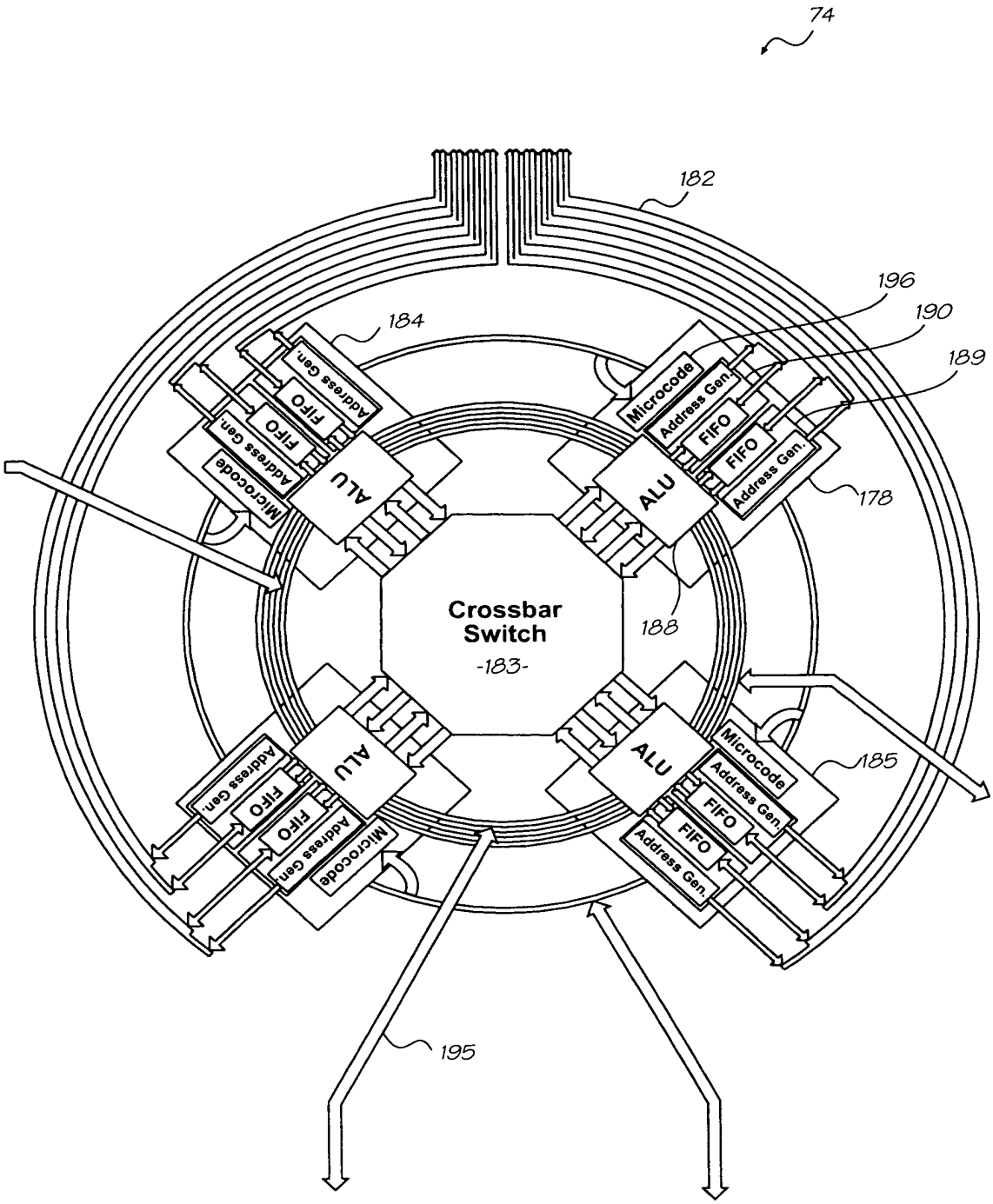

74
182
196
190
189
184
178
188
185
195
Crossbar Switch
-183-
ALU
Microcode
Address Gen.
FIFO
Address Gen.

FIG. 3(a)

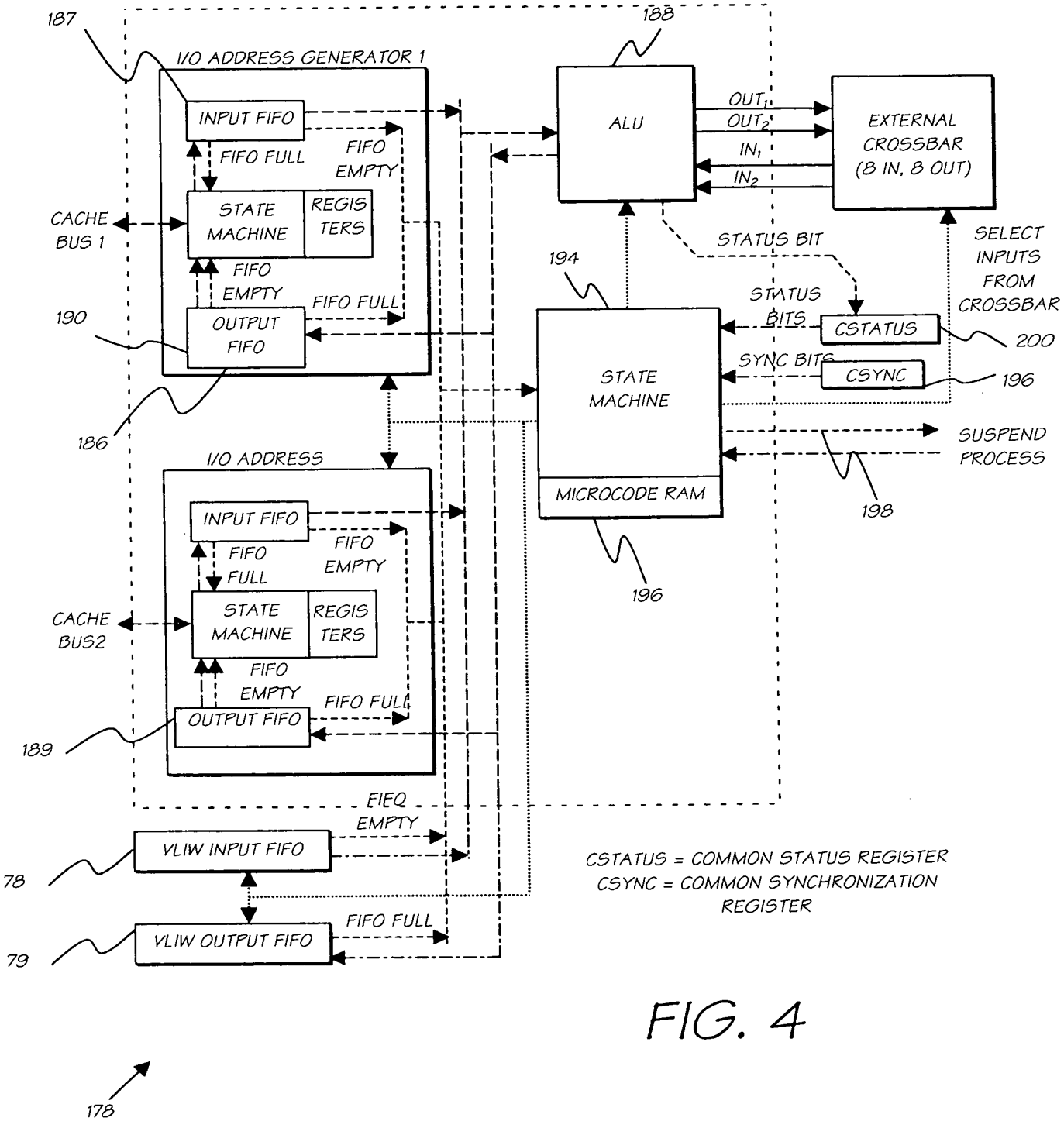

187
I/O ADDRESS GENERATOR 1
INPUT FIFO
FIFO FULL
FIFO EMPTY
CACHE BUS 1
STATE MACHINE
REGISTERS
FIFO EMPTY
FIFO FULL
190
OUTPUT FIFO
186
I/O ADDRESS
INPUT FIFO
FIFO FULL
FIFO EMPTY
CACHE BUS2
STATE MACHINE
REGISTERS
FIFO EMPTY
FIFO FULL
189
OUTPUT FIFO
188
ALU
OUT1
OUT2
IN1
IN2
EXTERNAL CROSSBAR (8 IN, 8 OUT)
STATUS BIT
SELECT INPUTS FROM CROSSBAR
194
STATUS BITS
CSTATUS
200
SYNC BITS
CSYNC
196
STATE MACHINE
SUSPEND PROCESS
198
MICROCODE RAM
196
FIFO EMPTY
VLIW INPUT FIFO
78
FIFO FULL
VLIW OUTPUT FIFO
79
CSTATUS = COMMON STATUS REGISTER
CSYNC = COMMON SYNCHRONIZATION REGISTER
178

FIG. 4

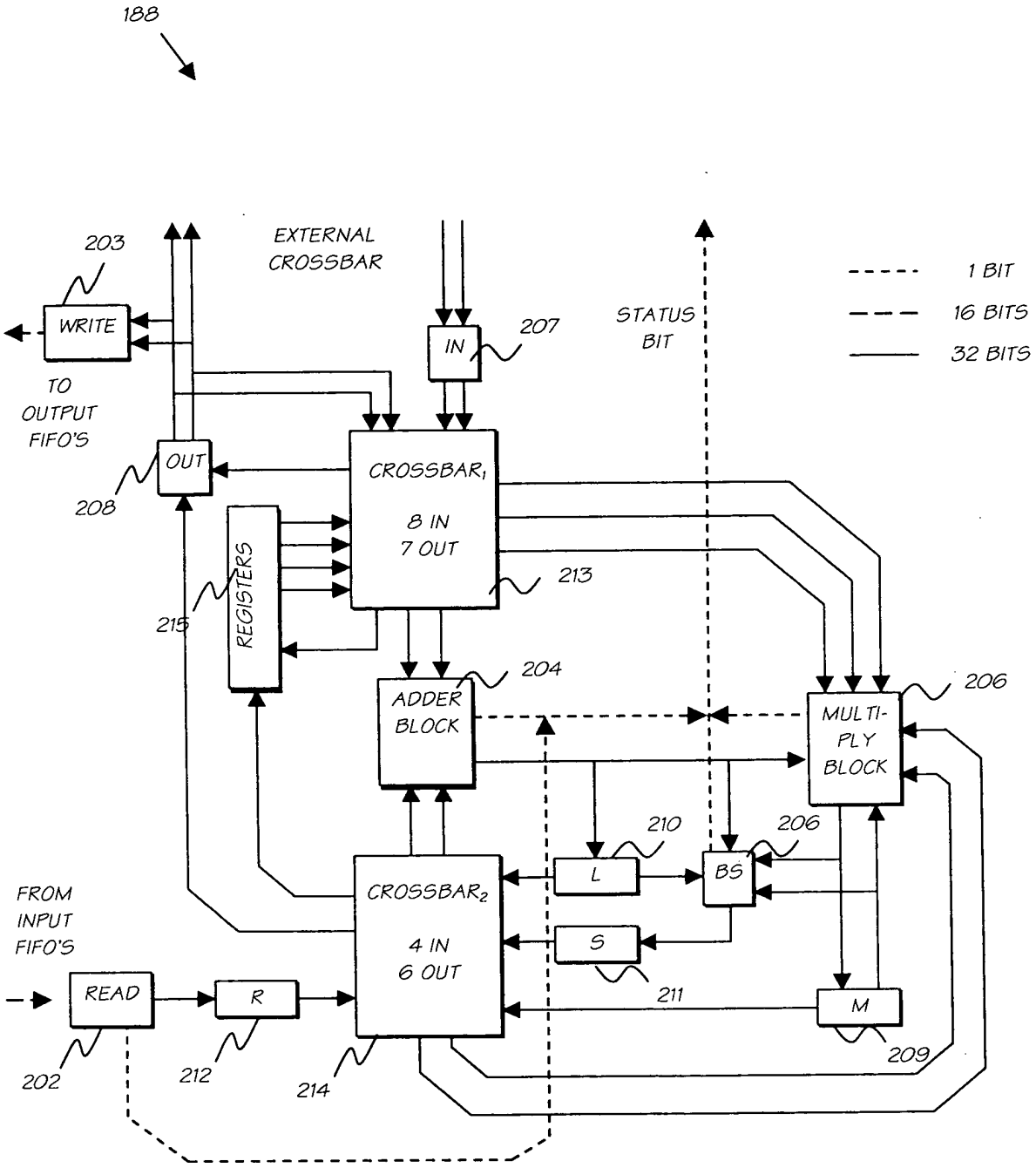

188
203
EXTERNAL CROSSBAR
WRITE
TO OUTPUT FIFO'S
OUT
208
IN
207
STATUS BIT
1 BIT
16 BITS
32 BITS
CROSSBAR1
8 IN
7 OUT
213
REGISTERS
215
ADDER BLOCK
204
MULTI-PLY BLOCK
206
210
206
L
BS
CROSSBAR2
4 IN
6 OUT
FROM INPUT FIFO'S
S
211
READ
R
M
202
212
214
209

FIG. 5

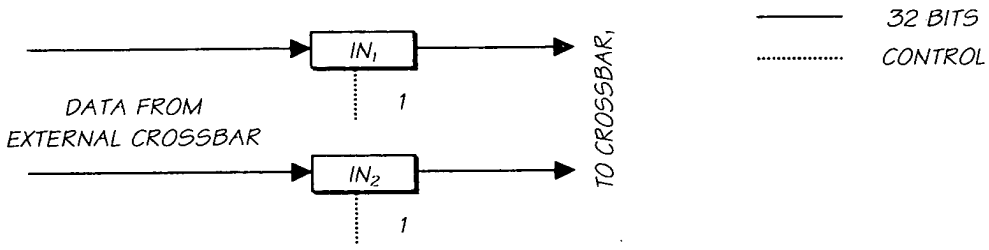

32 BITS
CONTROL
IN1
1
DATA FROM
EXTERNAL CROSSBAR
IN2
1
TO CROSSBAR1

FIG. 6

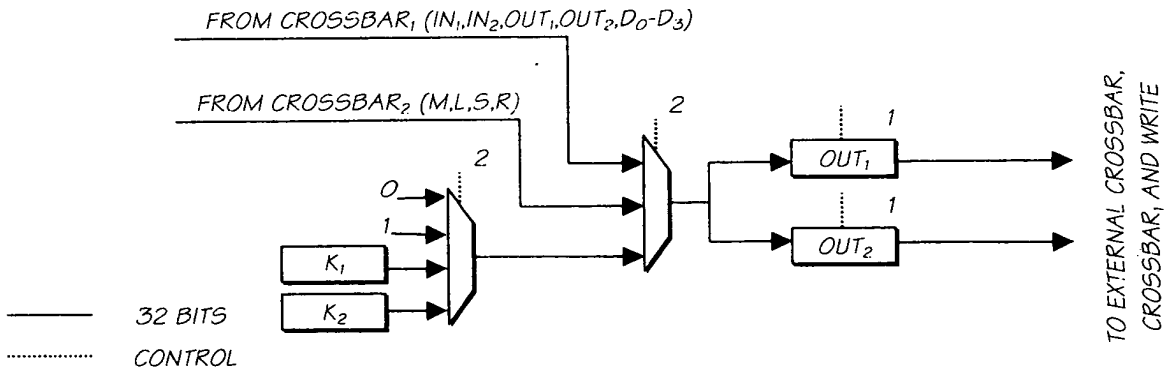

FROM CROSSBAR1 (IN1,IN2,OUT1,OUT2,D0-D3)
FROM CROSSBAR2 (M,L,S,R)
2
2
0
1
K1
K2
1
OUT1
1
OUT2
32 BITS
CONTROL
TO EXTERNAL CROSSBAR,
CROSSBAR, AND WRITE

FIG. 7

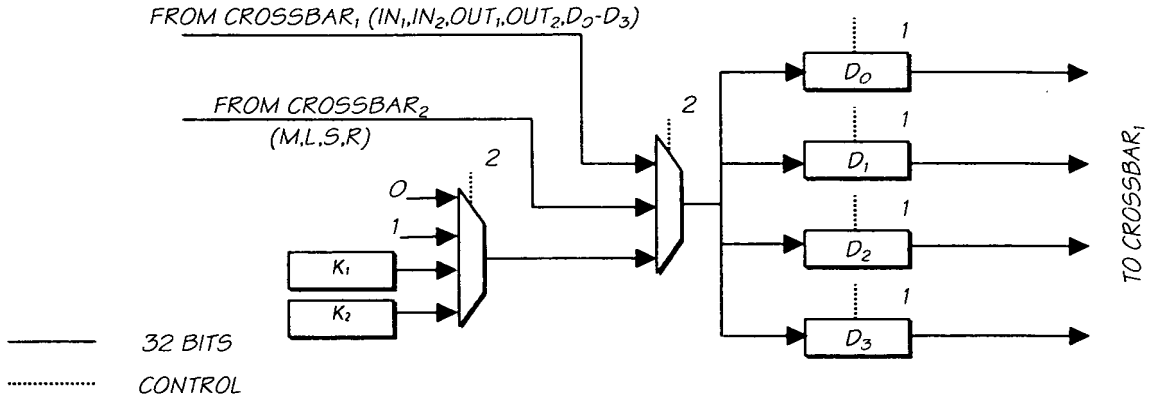

FROM CROSSBAR1 (IN1,IN2,OUT1,OUT2,D0-D3)
FROM CROSSBAR2
(M,L,S,R)
2
2
0
1
K1
K2
1
D0
1
D1
1
D2
1
D3
TO CROSSBAR1
32 BITS
CONTROL

FIG. 8

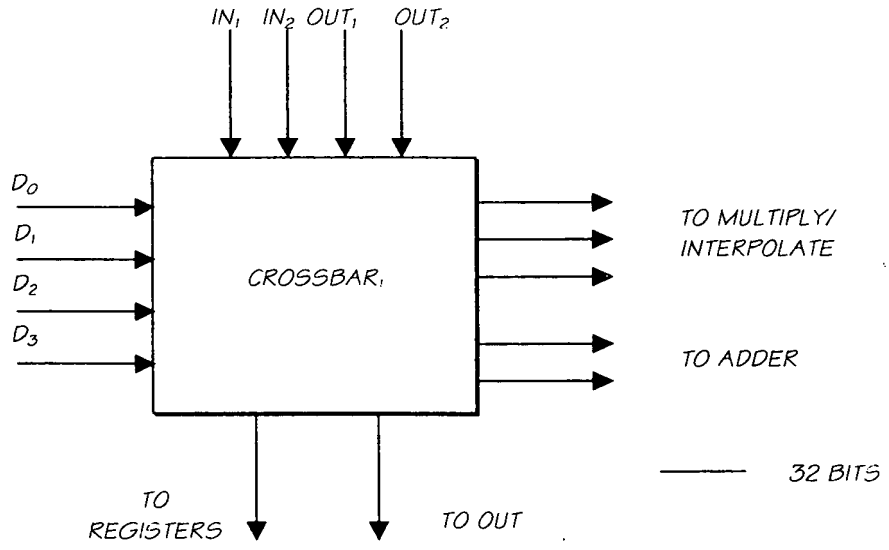

IN1
IN2
OUT1
OUT2
D0
D1
D2
D3
CROSSBAR1
TO MULTIPLY/
INTERPOLATE
TO ADDER
32 BITS
TO
REGISTERS
TO OUT

FIG. 9

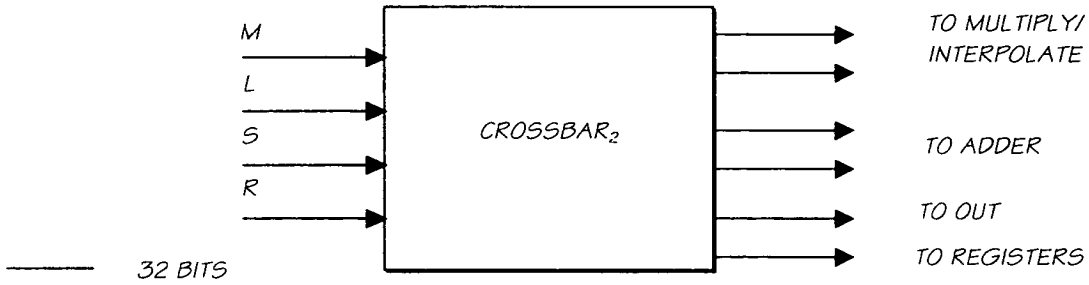

M
L
S
R
CROSSBAR2
TO MULTIPLY/
INTERPOLATE
TO ADDER
TO OUT
TO REGISTERS
32 BITS

FIG. 10

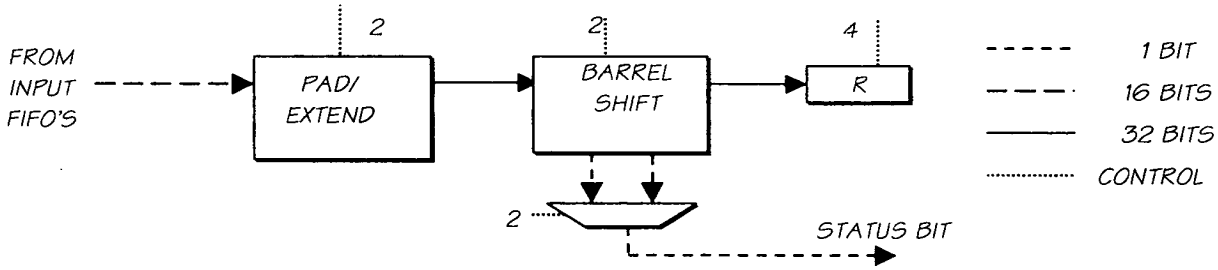

FROM
INPUT
FIFO'S
2
PAD/
EXTEND
2
BARREL
SHIFT
4
R
2
STATUS BIT
1 BIT
16 BITS
32 BITS
CONTROL

FIG. 11

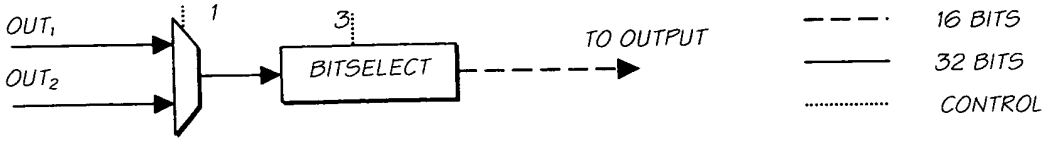


FIG. 12

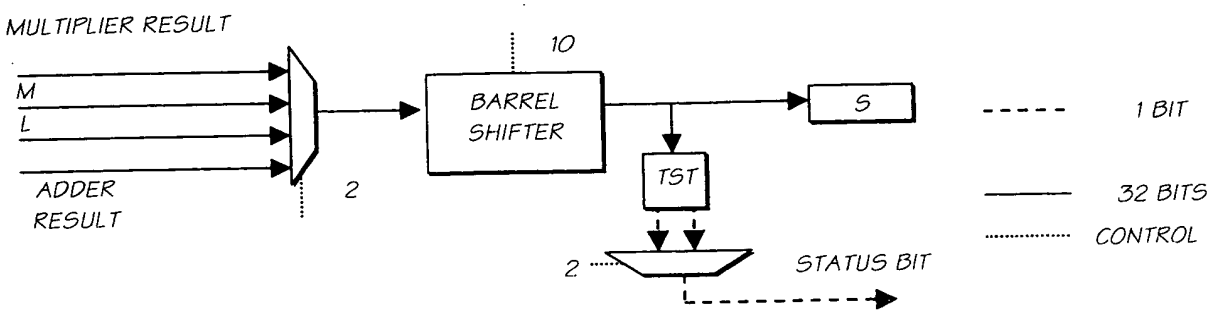


FIG. 13

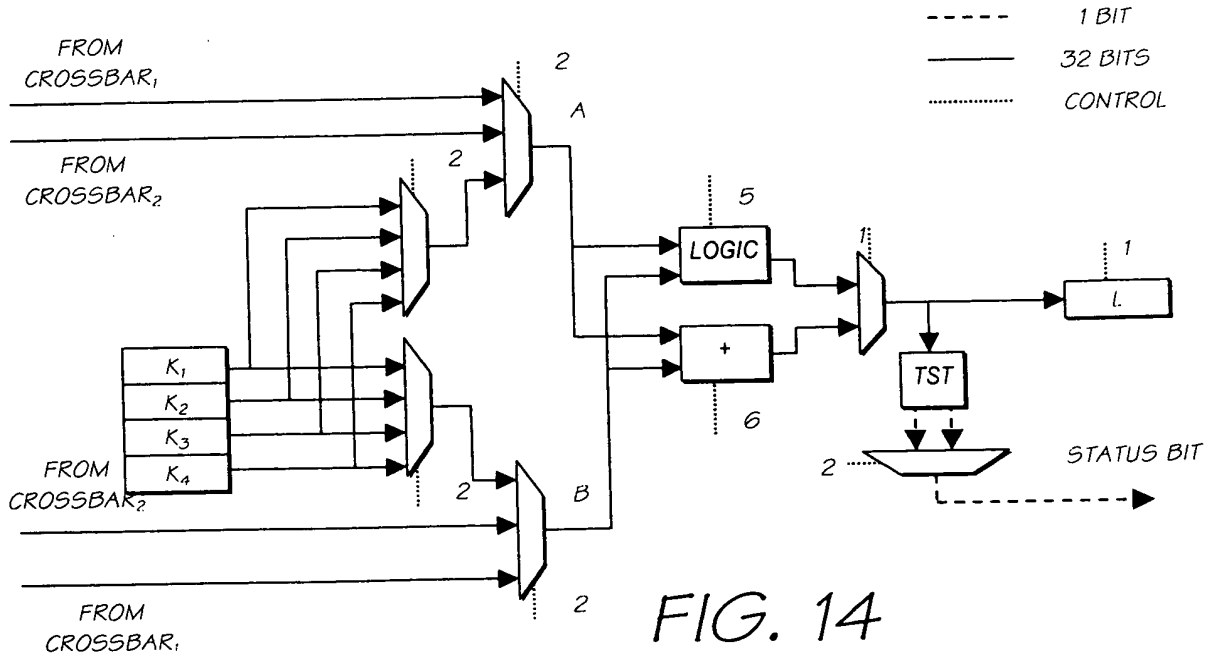


FIG. 14

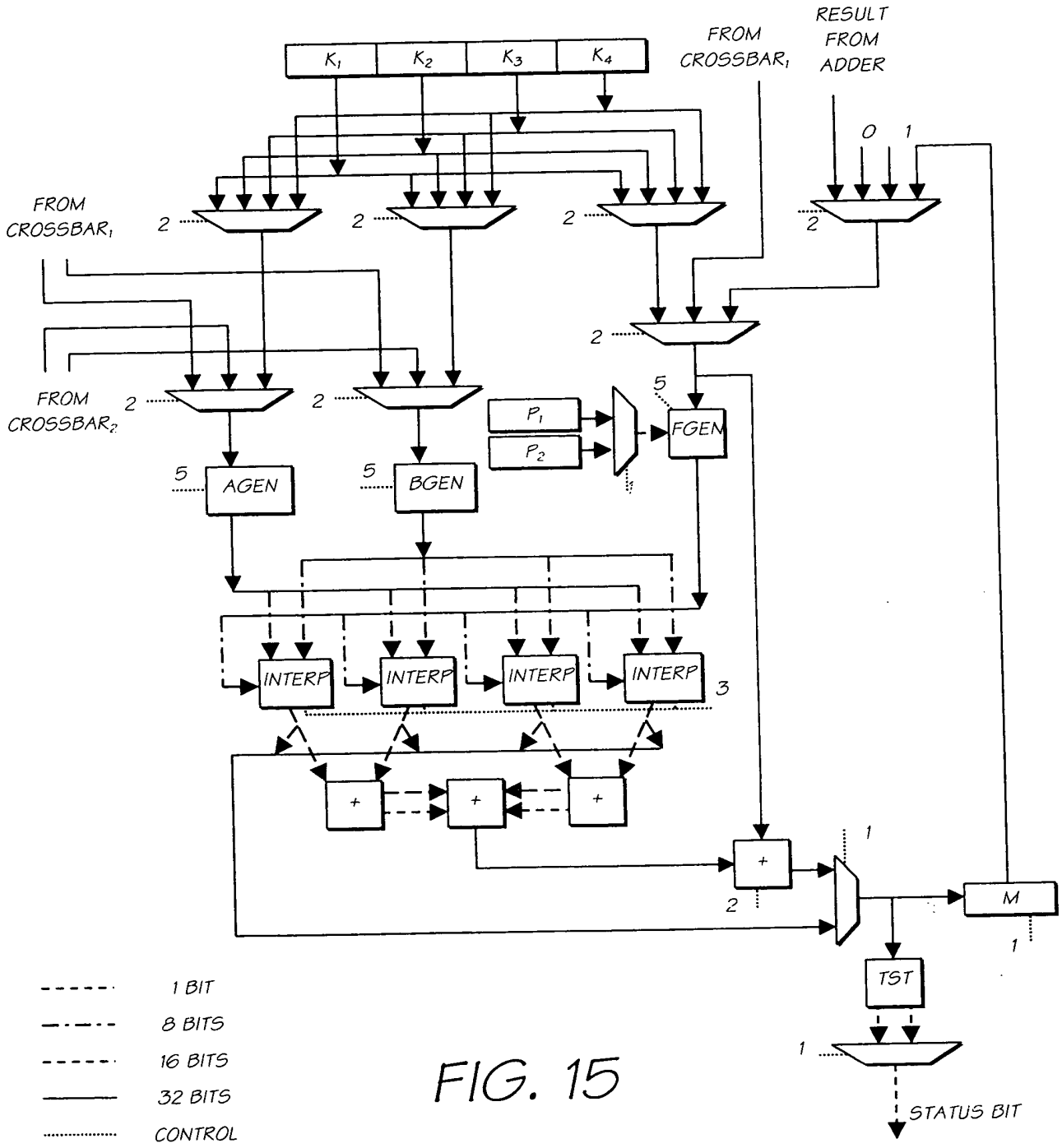

K1
K2
K3
K4
FROM CROSSBAR1
RESULT FROM ADDER
0
1
FROM CROSSBAR1
2
2
2
2
FROM CROSSBAR2
2
2
2
5
P1
P2
5
FGEN
5
AGEN
5
BGEN
INTERP
INTERP
INTERP
INTERP
3
+
+
+
+
2
1
M
1
TST
1
STATUS BIT
1 BIT
8 BITS
16 BITS
32 BITS
CONTROL

FIG. 15

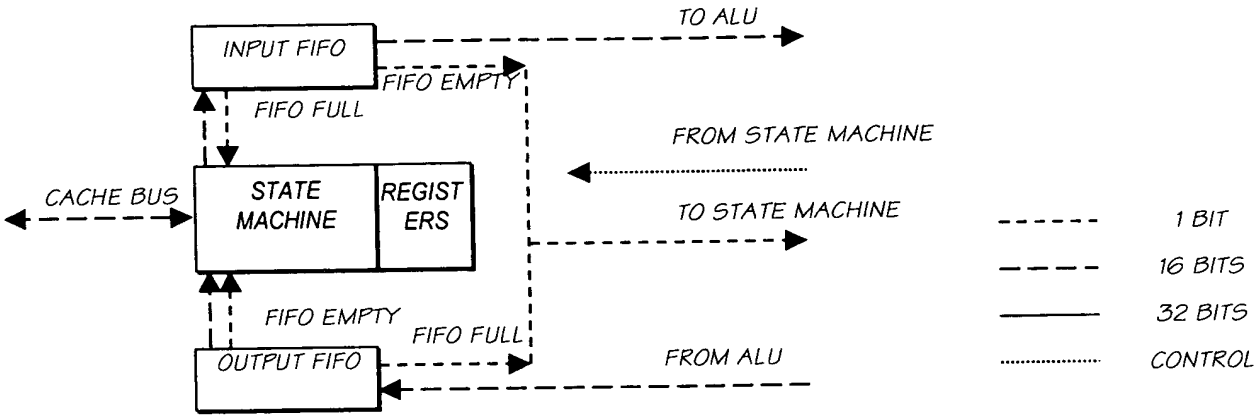


FIG. 16

ORDER OF PIXELS PRESENTED BY A SEQUENTIAL READ ITERATOR ON A 4 X 2 IMAGE WITH PADDING.

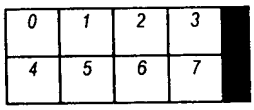


FIG. 17

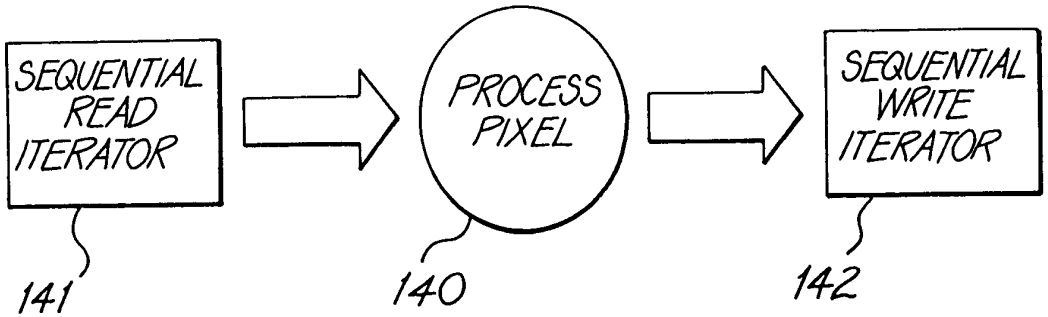
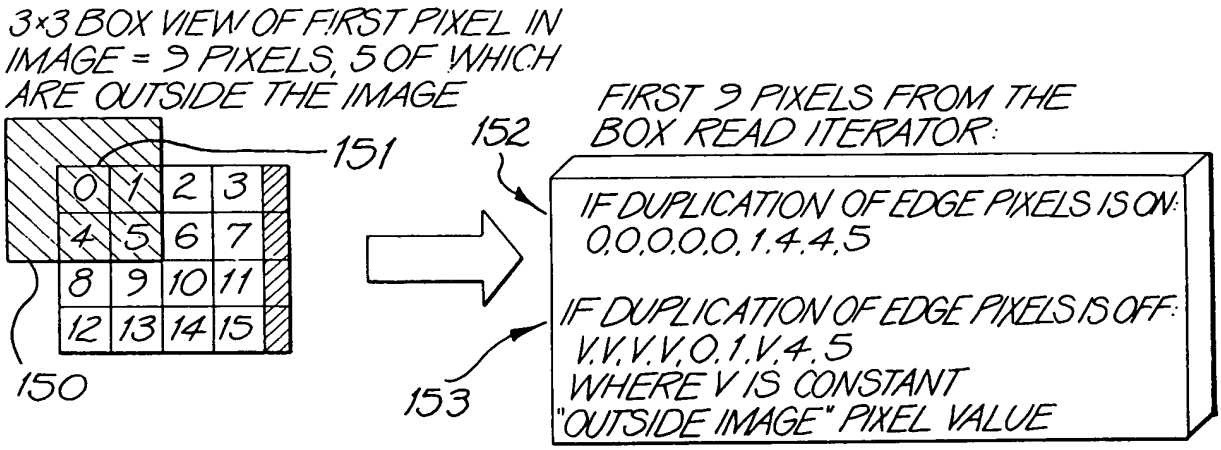

SEQUENTIAL READ ITERATOR
PROCESS PIXEL
SEQUENTIAL WRITE ITERATOR
141
140
142

FIG. 18

A 3×3 BOX VIEW TRAVERSES THE PIXELS IN ORDER: 0,1,2,3,4,5,6,7,8 ETC, PLACING A 3×3 BOX CENTERED OVER EACH PIXEL...



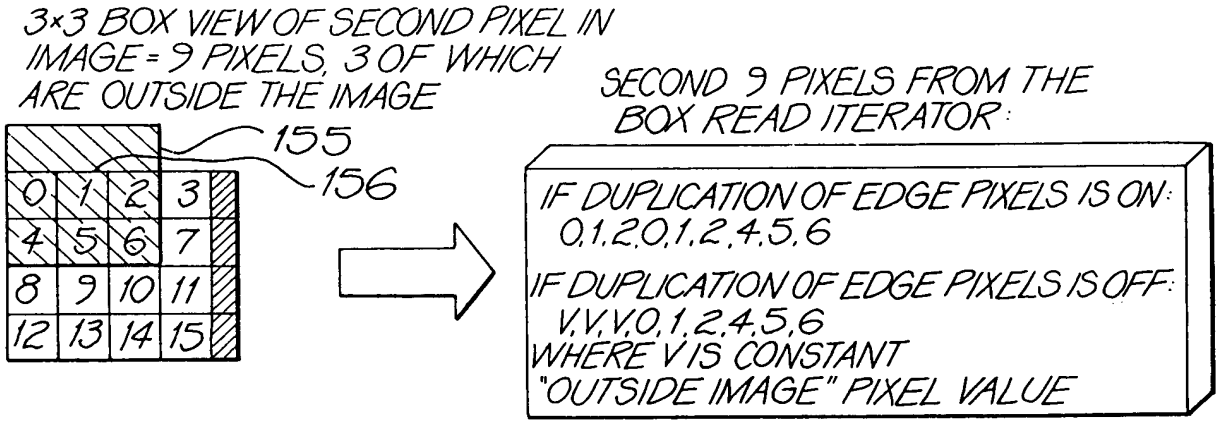


FIG. 19

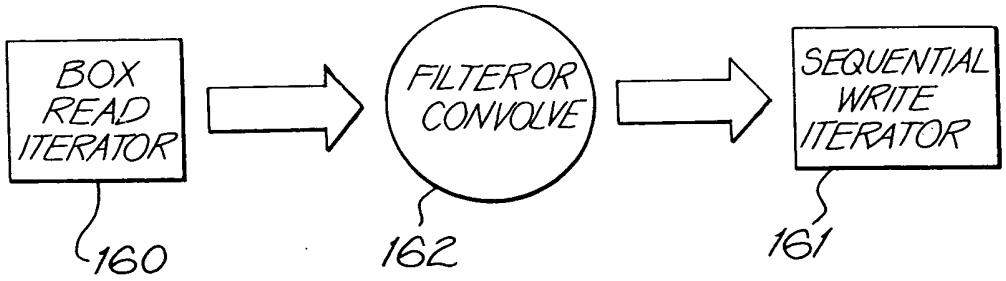

BOX READ ITERATOR
FILTER OR CONVOLVE
SEQUENTIAL WRITE ITERATOR
160
162
161

FIG. 20

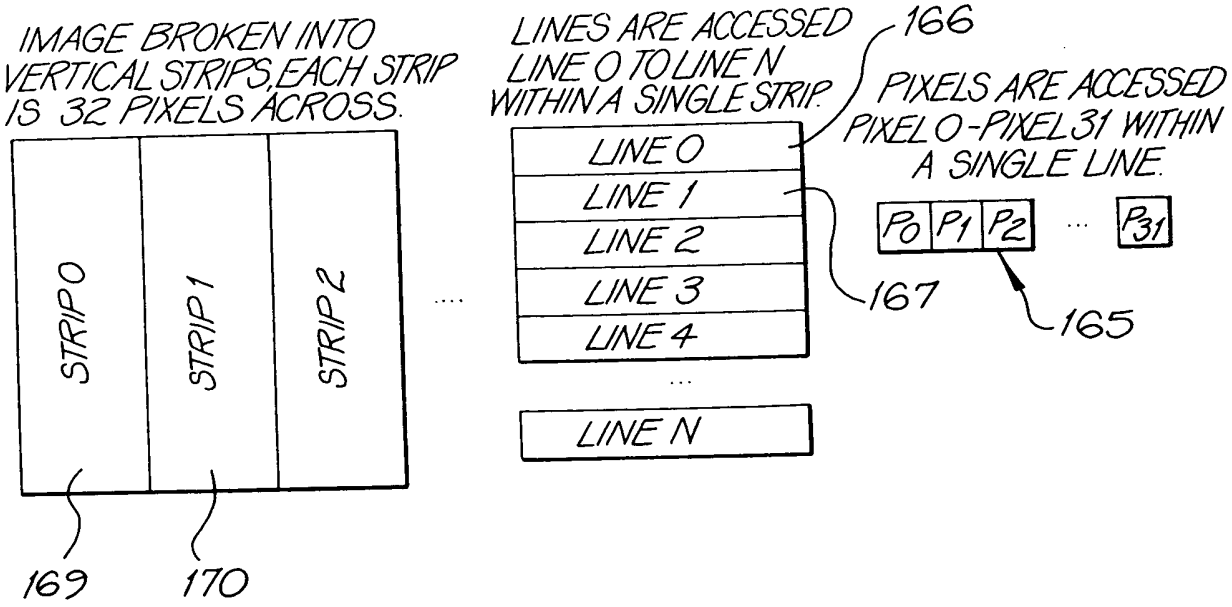

IMAGE BROKEN INTO VERTICAL STRIPS, EACH STRIP IS 32 PIXELS ACROSS.
STRIP 0
STRIP 1
STRIP 2
169
170
LINES ARE ACCESSED LINE 0 TO LINE N WITHIN A SINGLE STRIP.
166
LINE 0
LINE 1
LINE 2
LINE 3
LINE 4
LINE N
167
PIXELS ARE ACCESSED PIXEL 0 - PIXEL 31 WITHIN A SINGLE LINE.
P0 P1 P2 ... P31
165

FIG. 21

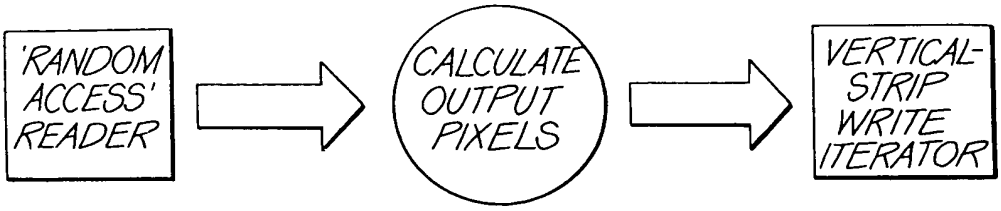

'RANDOM ACCESS' READER
CALCULATE OUTPUT PIXELS
VERTICAL-STRIP WRITE ITERATOR

FIG. 22

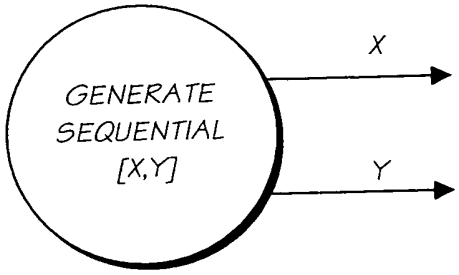


FIG. 23

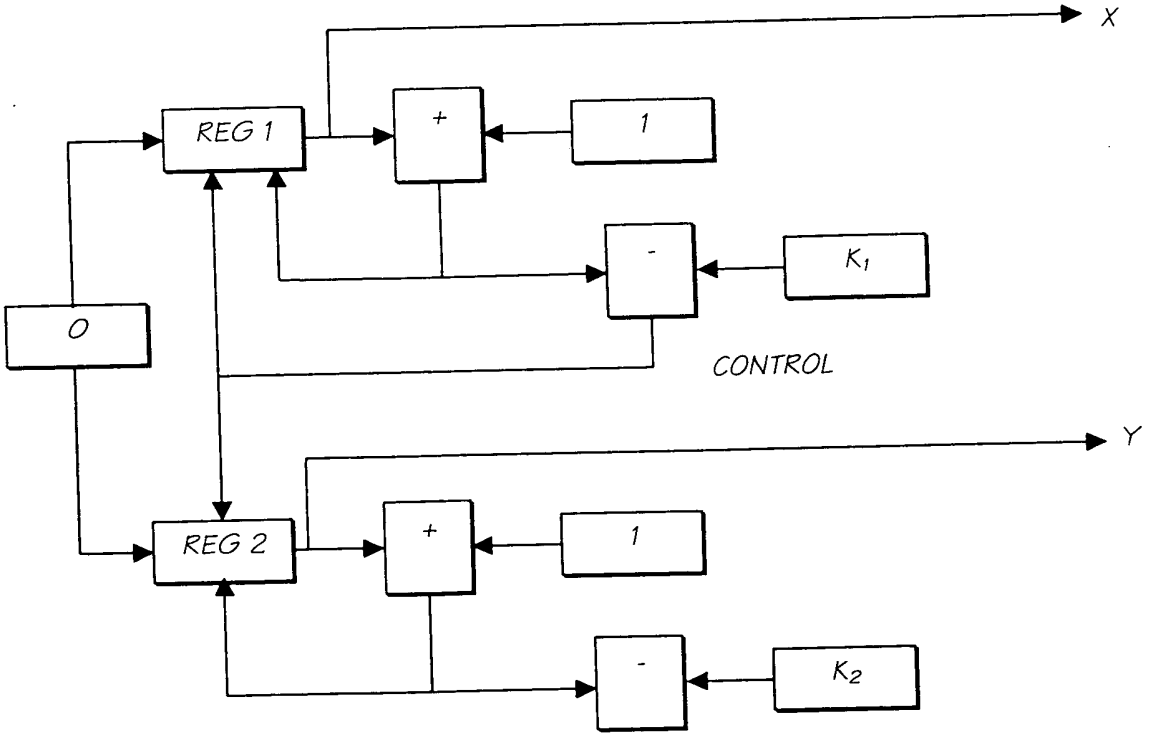


FIG. 24

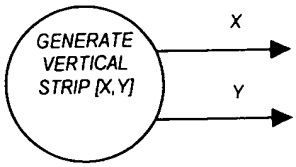

GENERATE
VERTICAL
STRIP [X,Y]
X
Y

FIG. 25

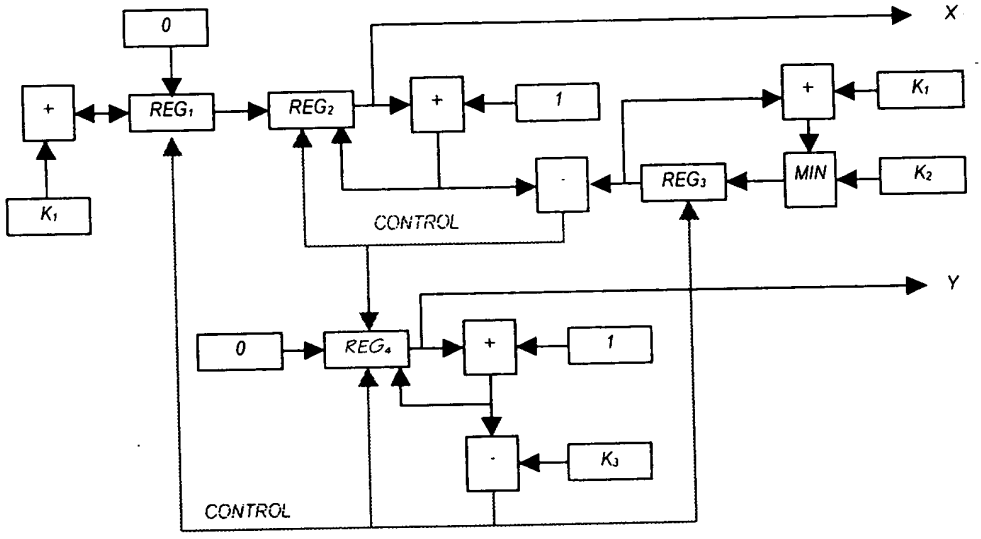
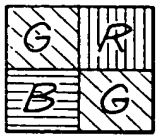

0
REG1
REG2
+
1
X
+
K1
MIN
K2
REG3
-
K1
CONTROL
Y
0
REG4
+
1
-
K3
CONTROL

FIG. 26



2×2 PIXEL BLOCK FROM CCD

FIG. 27

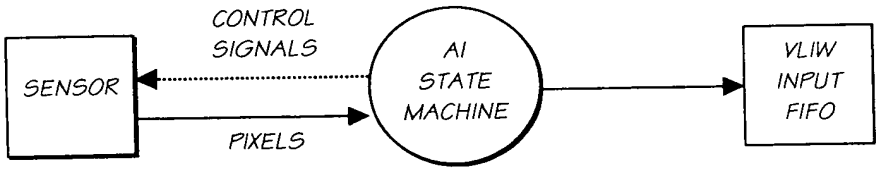


FIG. 28

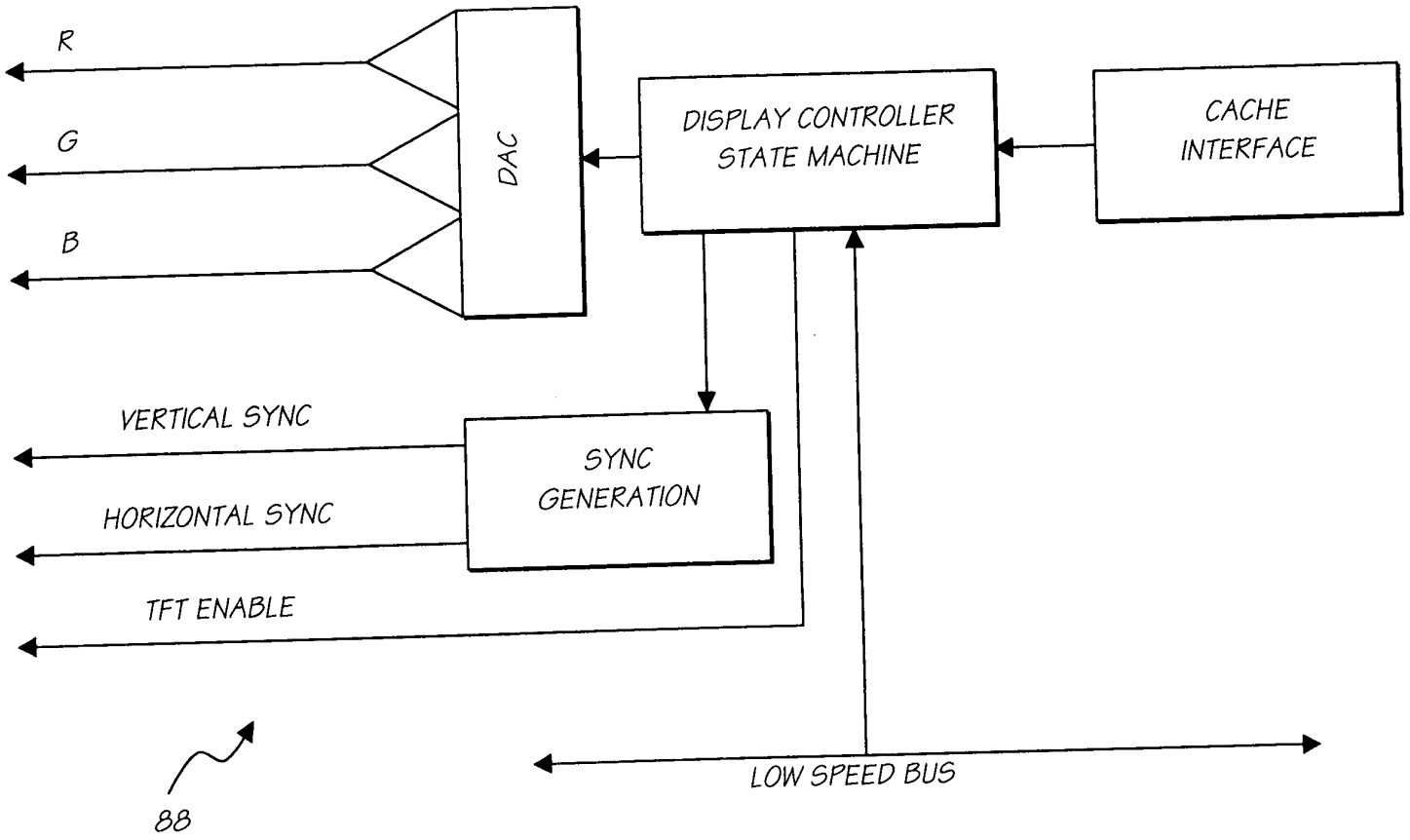
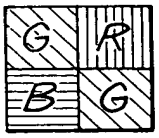

R
G
B
DAC
DISPLAY CONTROLLER STATE MACHINE
CACHE INTERFACE
VERTICAL SYNC
HORIZONTAL SYNC
SYNC GENERATION
TFT ENABLE
LOW SPEED BUS
88

FIG. 29



2x2 PIXEL BLOCK FROM CCD

FIG. 30

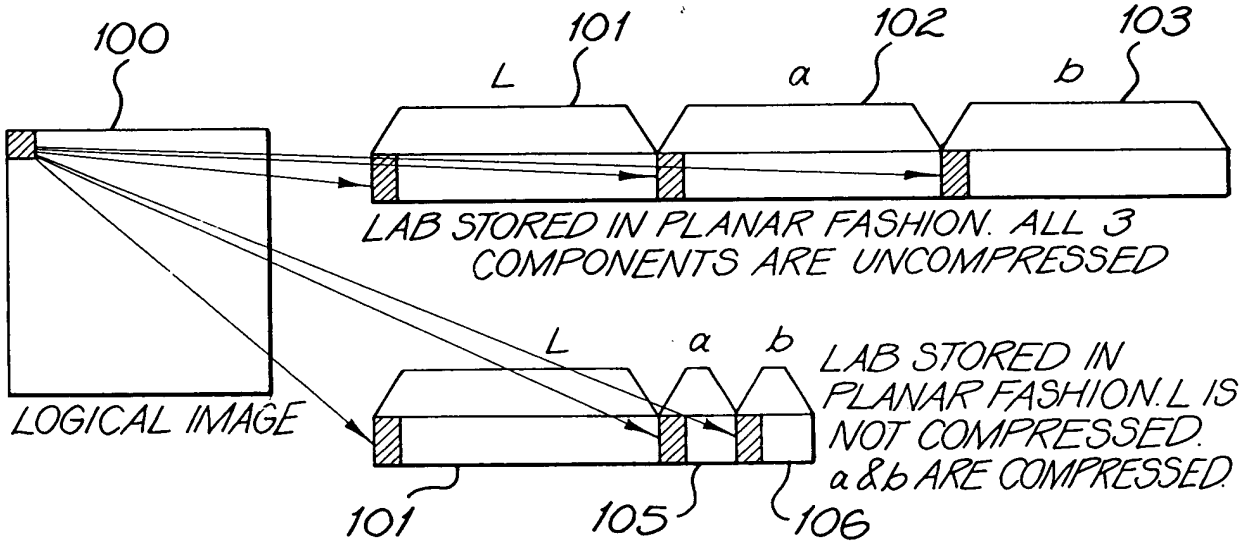


FIG. 31

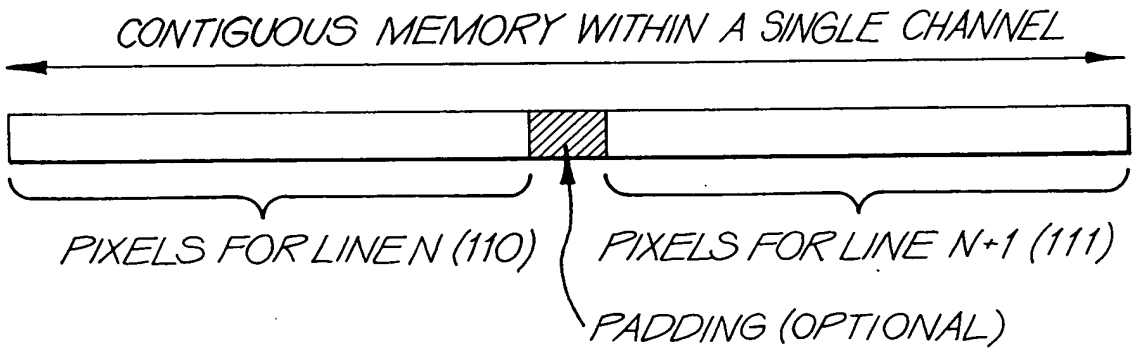


FIG. 32

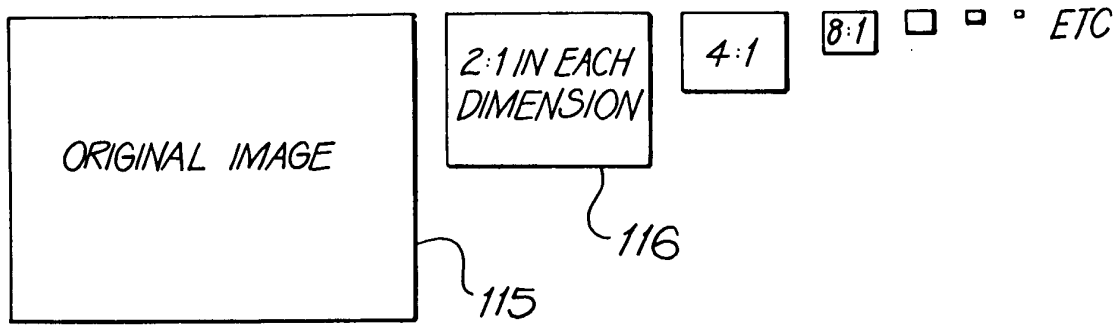


FIG. 33

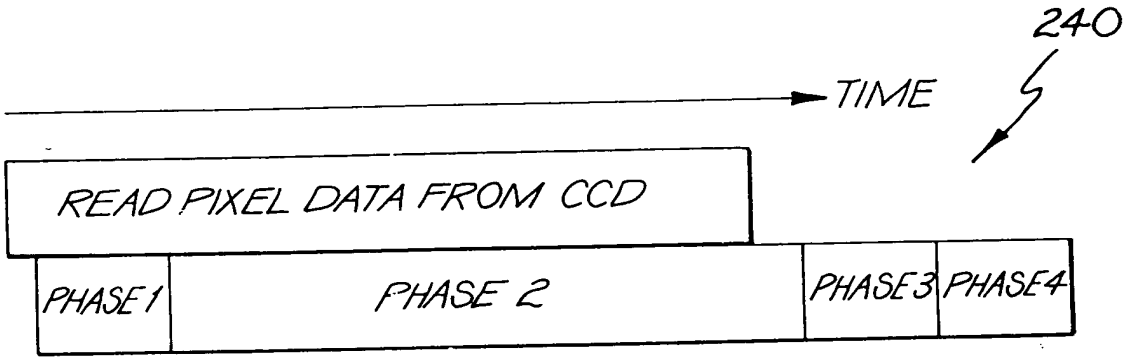

240
TIME
READ PIXEL DATA FROM CCD
PHASE 1
PHASE 2
PHASE 3
PHASE 4

FIG. 34

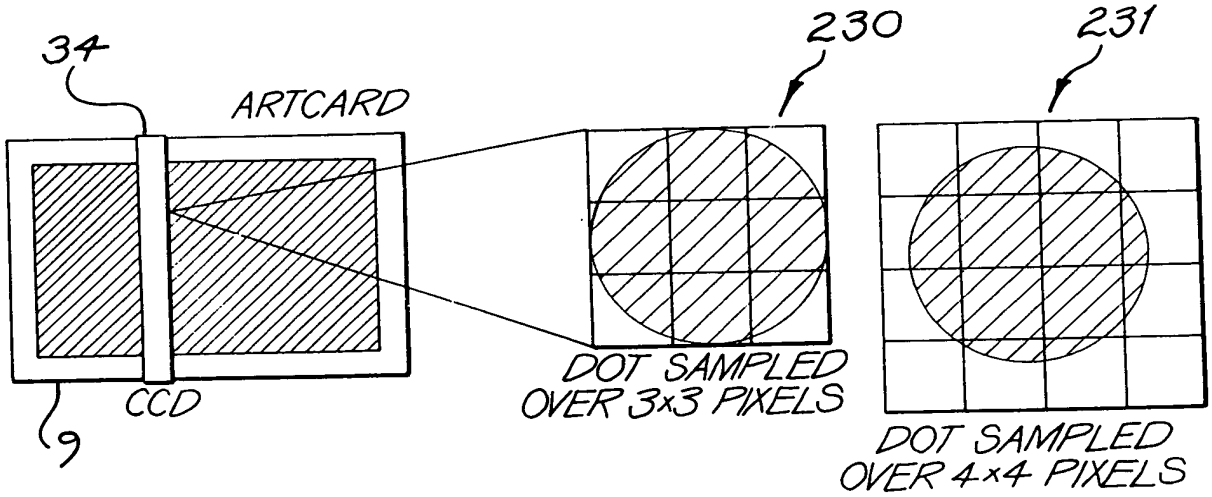

34
ARTCARD
CCD
9
230
DOT SAMPLED
OVER 3×3 PIXELS
231
DOT SAMPLED
OVER 4×4 PIXELS

FIG. 35

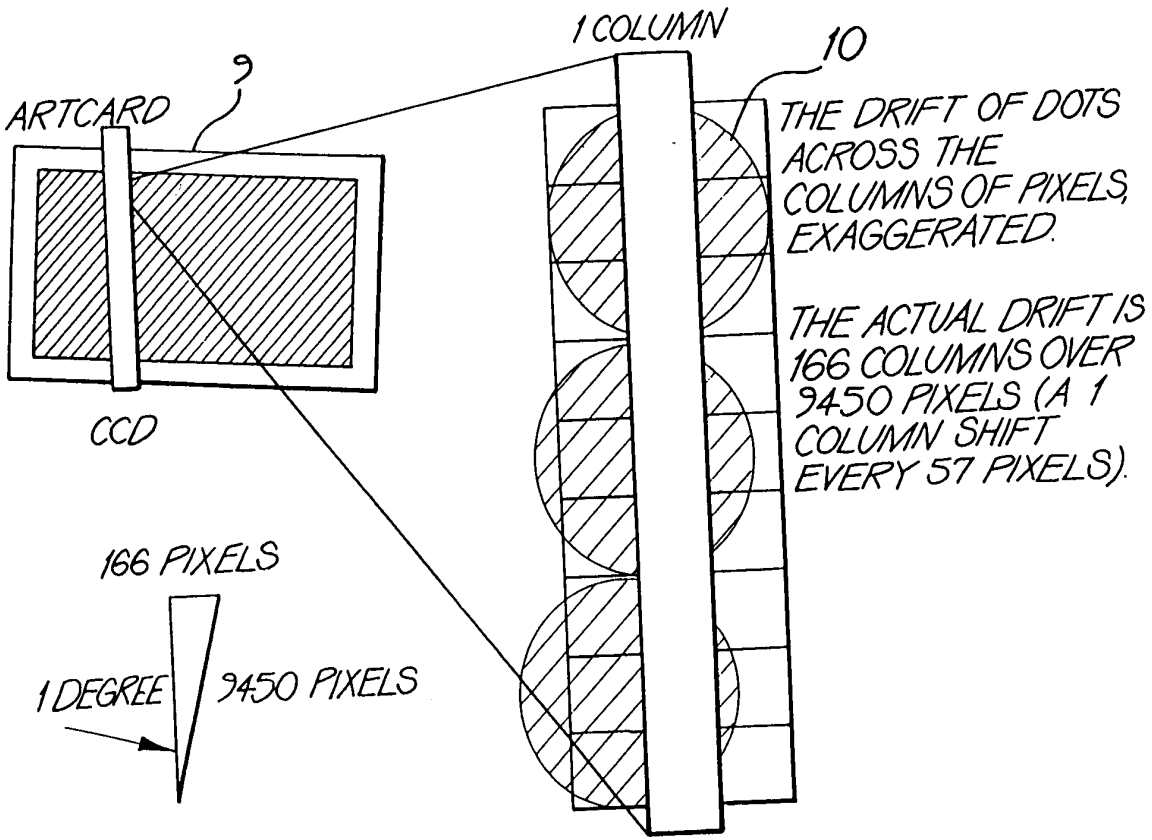

1 COLUMN
10
ARTCARD
9
THE DRIFT OF DOTS ACROSS THE COLUMNS OF PIXELS, EXAGGERATED.
THE ACTUAL DRIFT IS 166 COLUMNS OVER 9450 PIXELS (A 1 COLUMN SHIFT EVERY 57 PIXELS).
CCD
166 PIXELS
1 DEGREE
9450 PIXELS

FIG. 36

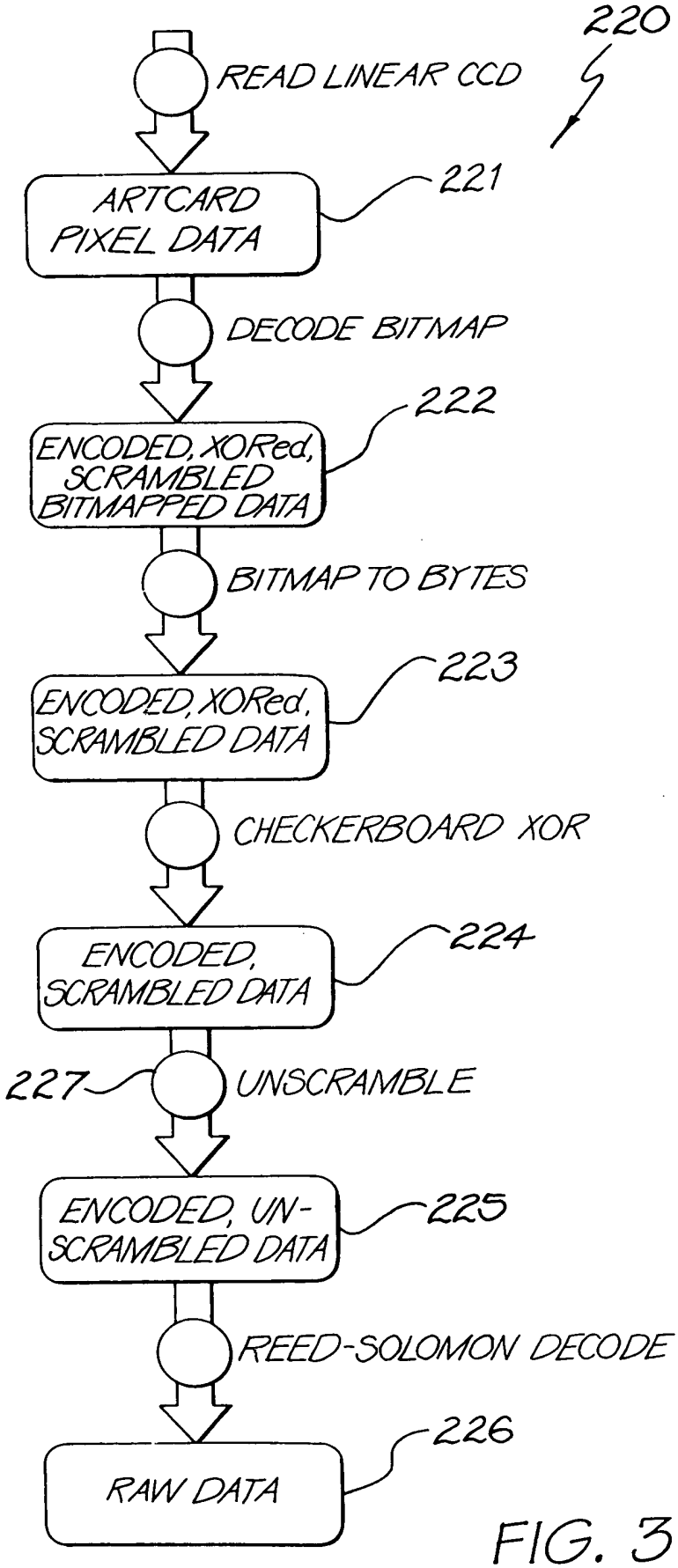

220
READ LINEAR CCD
ARTCARD PIXEL DATA
221
DECODE BITMAP
ENCODED, XORed, SCRAMBLED BITMAPPED DATA
222
BITMAP TO BYTES
ENCODED, XORed, SCRAMBLED DATA
223
CHECKERBOARD XOR
ENCODED, SCRAMBLED DATA
224
227
UNSCRAMBLE
ENCODED, UN-SCRAMBLED DATA
225
REED-SOLOMON DECODE
RAW DATA
226

FIG. 37

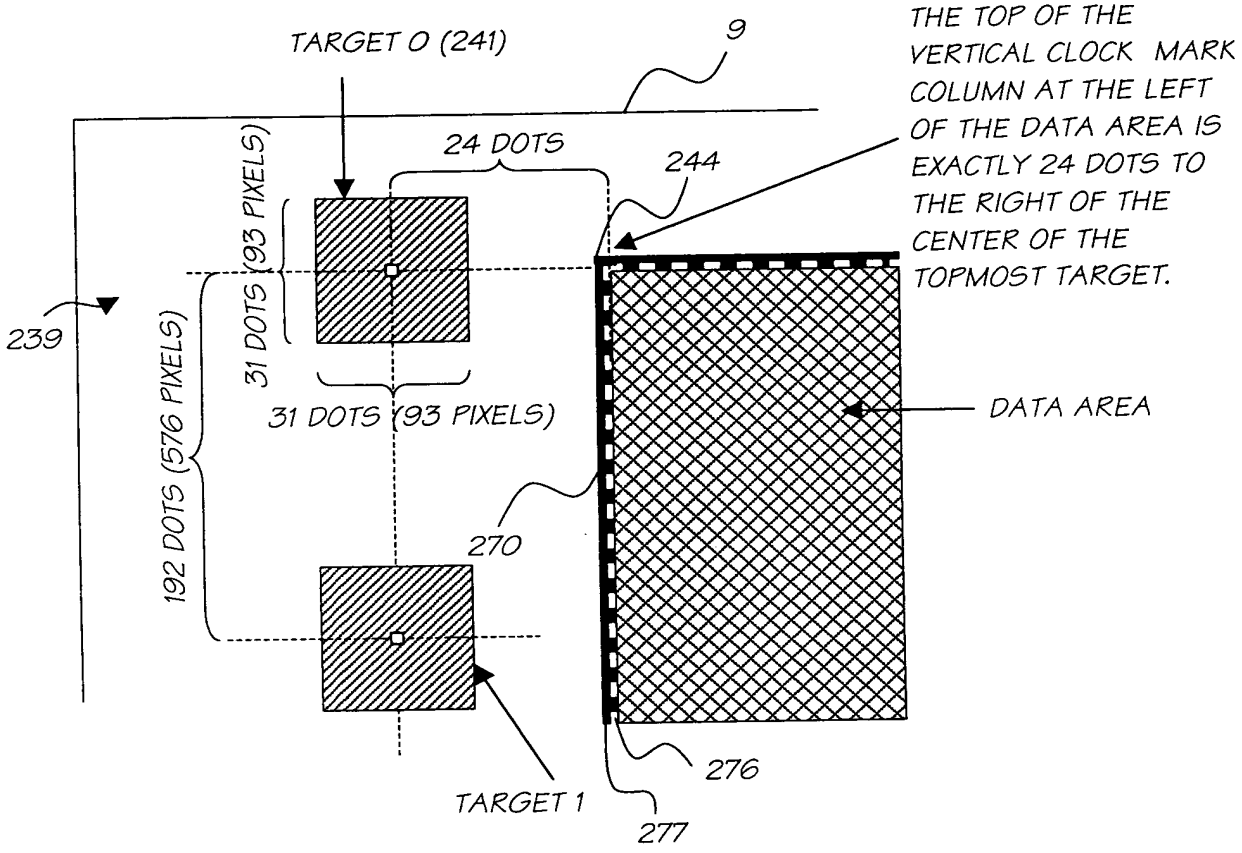

TARGET 0 (241)
9
THE TOP OF THE VERTICAL CLOCK MARK COLUMN AT THE LEFT OF THE DATA AREA IS EXACTLY 24 DOTS TO THE RIGHT OF THE CENTER OF THE TOPMOST TARGET.
24 DOTS
244
31 DOTS (93 PIXELS)
239
31 DOTS (93 PIXELS)
192 DOTS (576 PIXELS)
DATA AREA
270
276
TARGET 1
277

FIG. 38

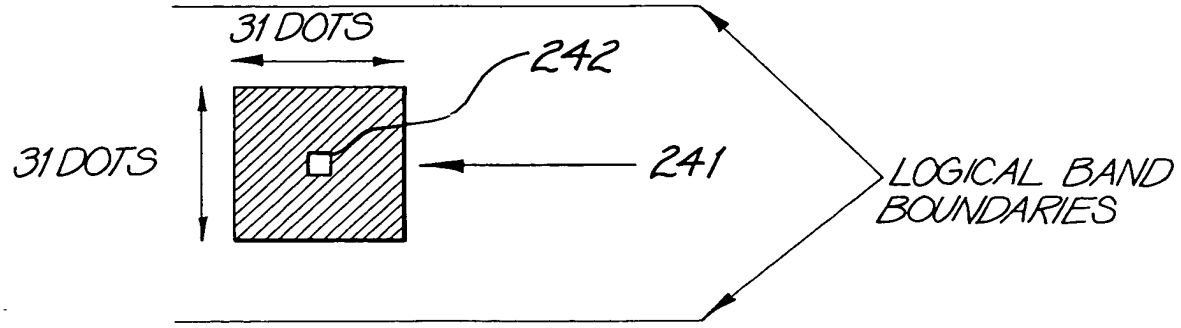

31 DOTS
242
31 DOTS
241
LOGICAL BAND BOUNDARIES

FIG. 39

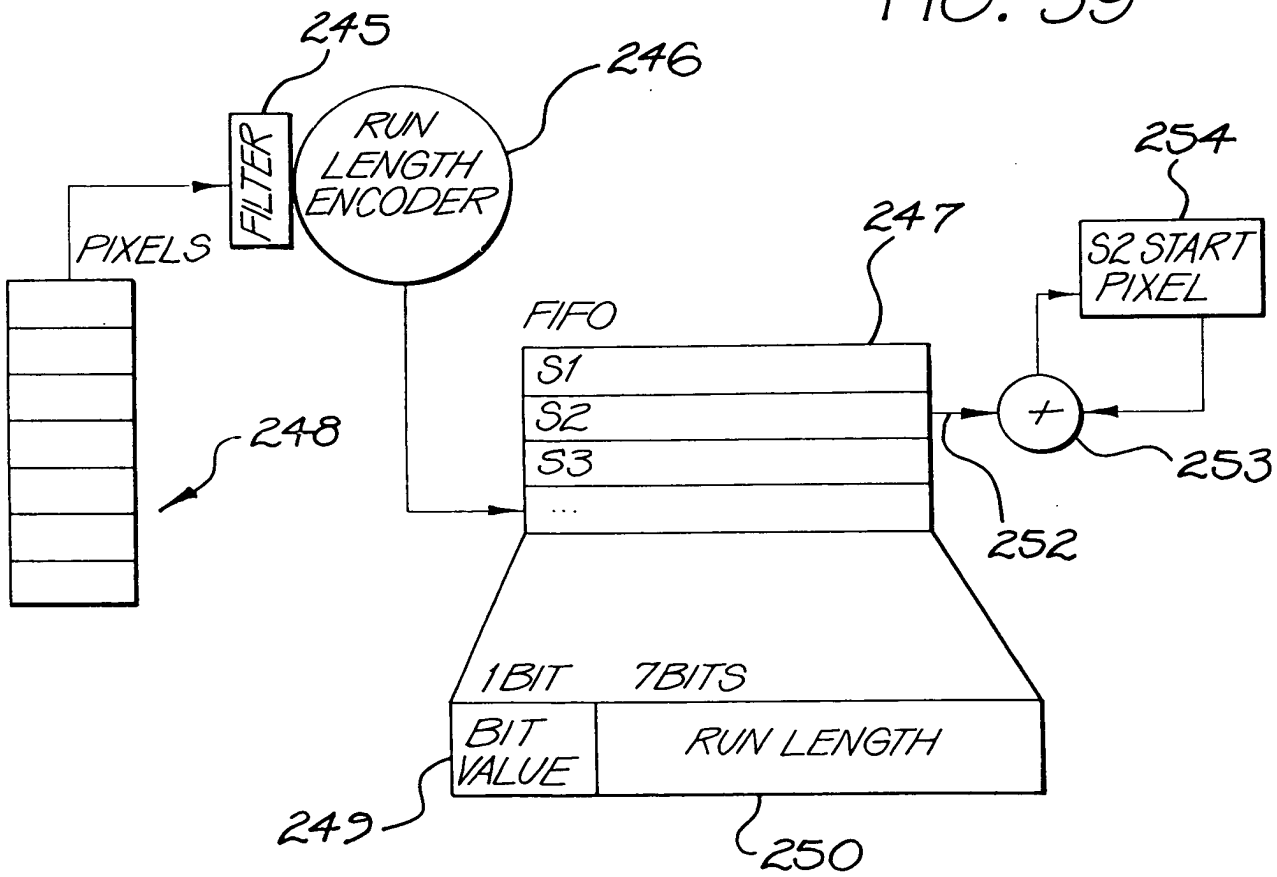

245
246
RUN LENGTH ENCODER
FILTER
PIXELS
254
S2 START PIXEL
247
FIFO
S1
S2
S3
...
248
253
252
1BIT
7BITS
BIT VALUE
RUN LENGTH
249
250

FIG. 40

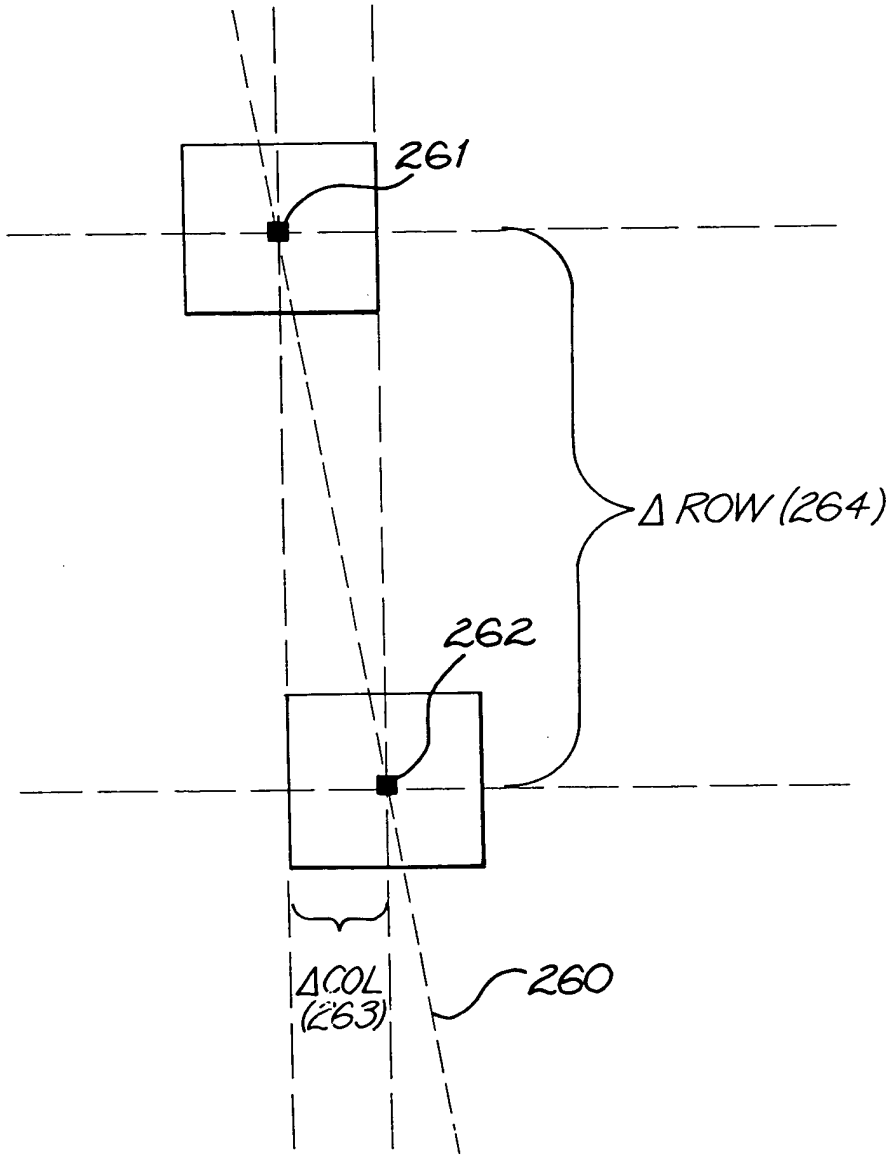


FIG. 41

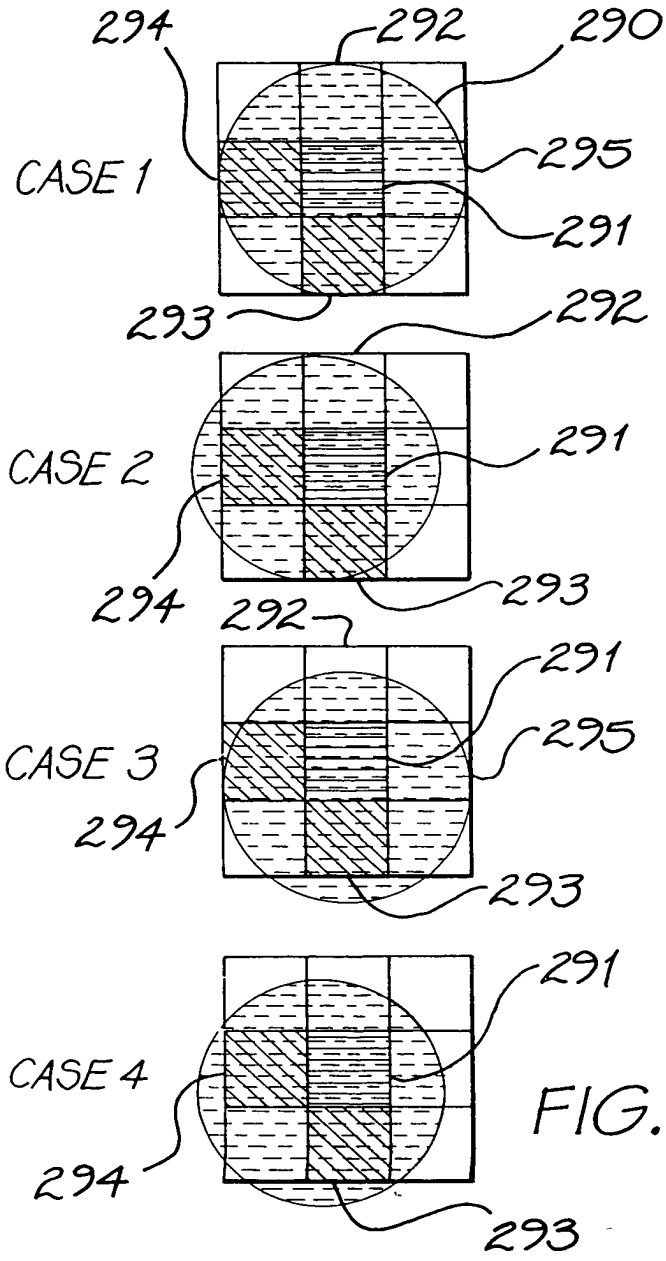

294
292
290
CASE 1
295
291
293
292
CASE 2
291
294
293
292
291
CASE 3
295
294
293
291
CASE 4
294
293

FIG. 42

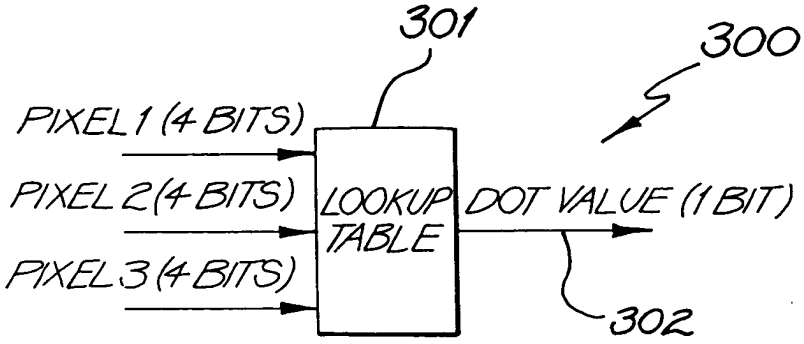

301
300
PIXEL 1 (4 BITS)
PIXEL 2 (4 BITS)
PIXEL 3 (4 BITS)
LOOKUP TABLE
DOT VALUE (1 BIT)
302

FIG. 43

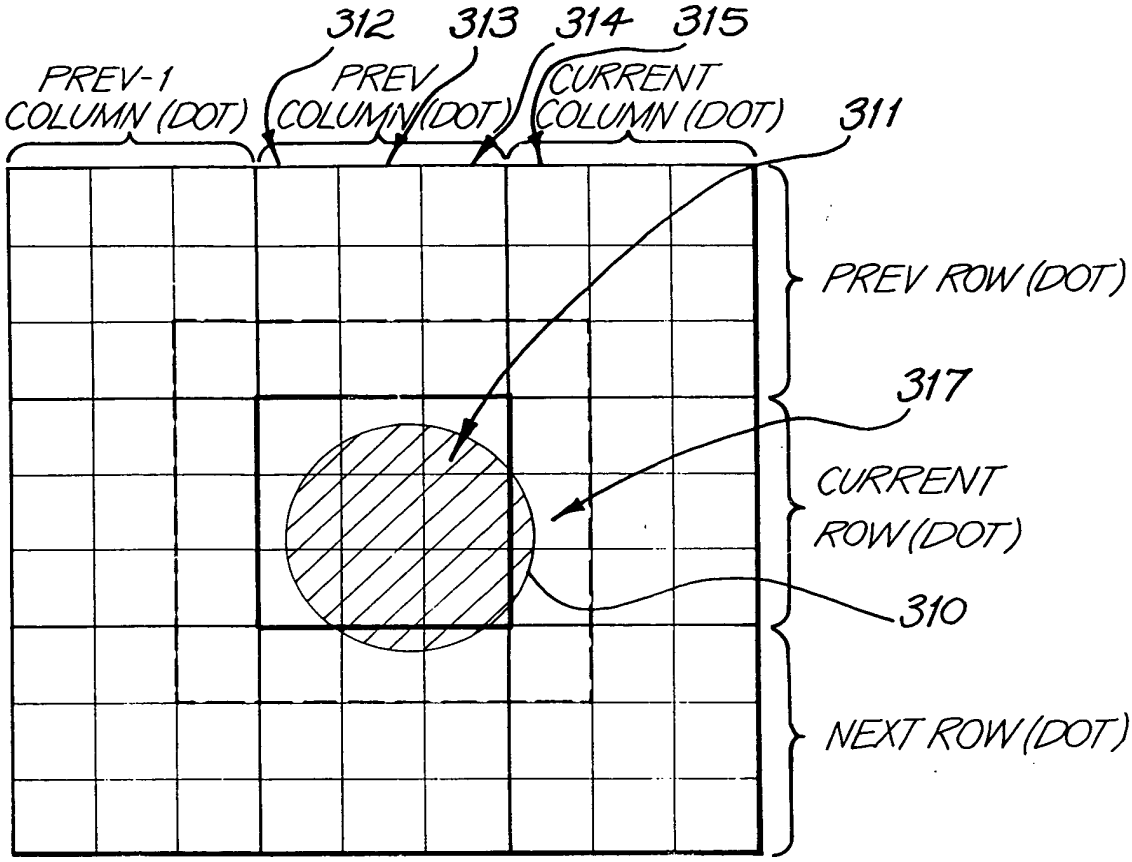

312
313
314
315
PREV-1 COLUMN (DOT)
PREV COLUMN (DOT)
CURRENT COLUMN (DOT)
311
PREV ROW (DOT)
317
CURRENT ROW (DOT)
310
NEXT ROW (DOT)

FIG. 44

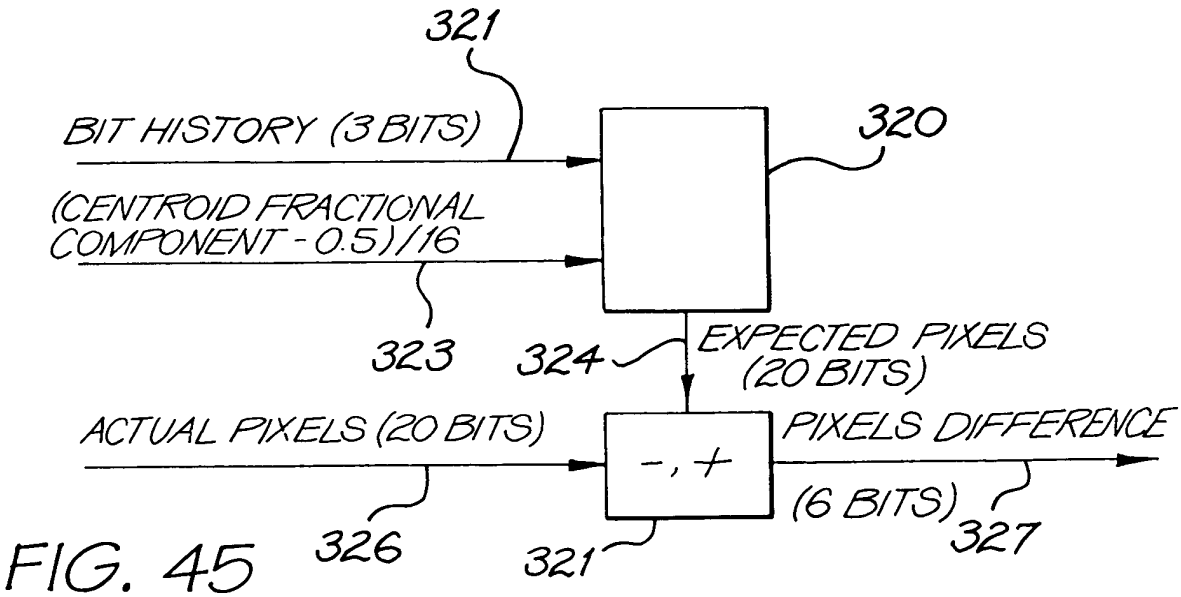

321
BIT HISTORY (3 BITS)
320
(CENTROID FRACTIONAL COMPONENT - 0.5)/16
323
324
EXPECTED PIXELS (20 BITS)
ACTUAL PIXELS (20 BITS)
-, +
PIXELS DIFFERENCE
(6 BITS)
326
321
327

FIG. 45

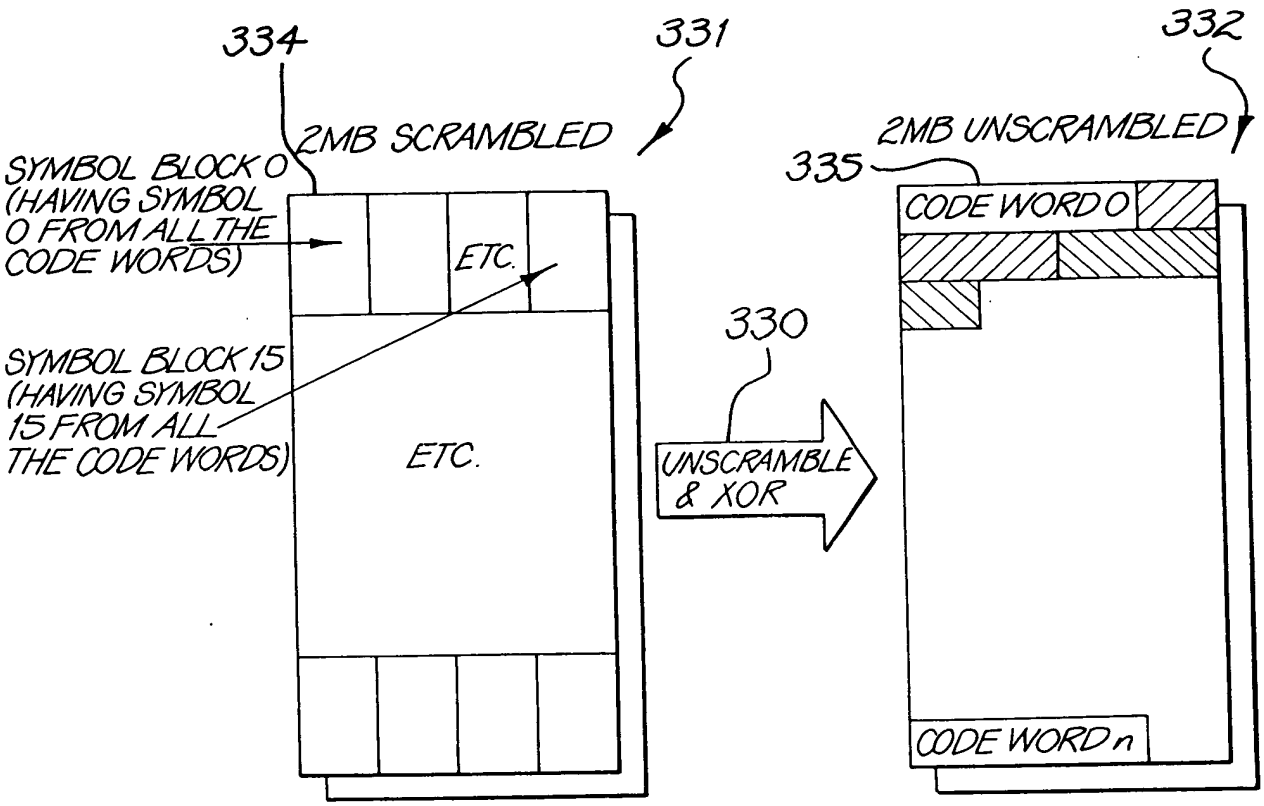

334
331
332
2MB SCRAMBLED
2MB UNSCRAMBLED
335
SYMBOL BLOCK 0 (HAVING SYMBOL 0 FROM ALL THE CODE WORDS)
ETC.
SYMBOL BLOCK 15 (HAVING SYMBOL 15 FROM ALL THE CODE WORDS)
ETC.
330
UNSCRAMBLE & XOR
CODE WORD 0
CODE WORD n

FIG. 46

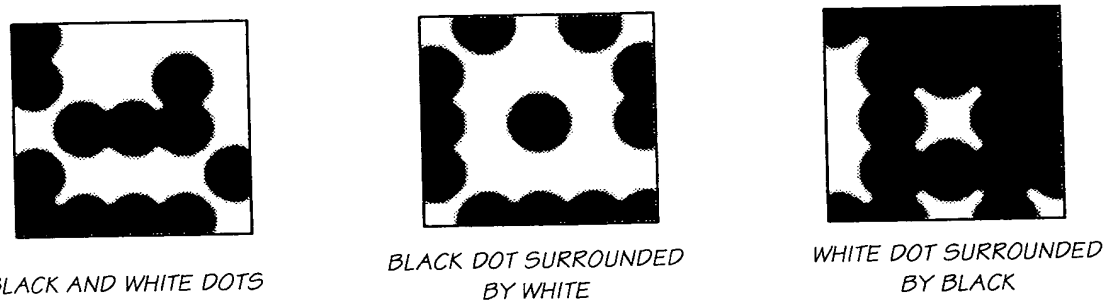

BLACK AND WHITE DOTS
BLACK DOT SURROUNDED BY WHITE
WHITE DOT SURROUNDED BY BLACK

FIG. 47

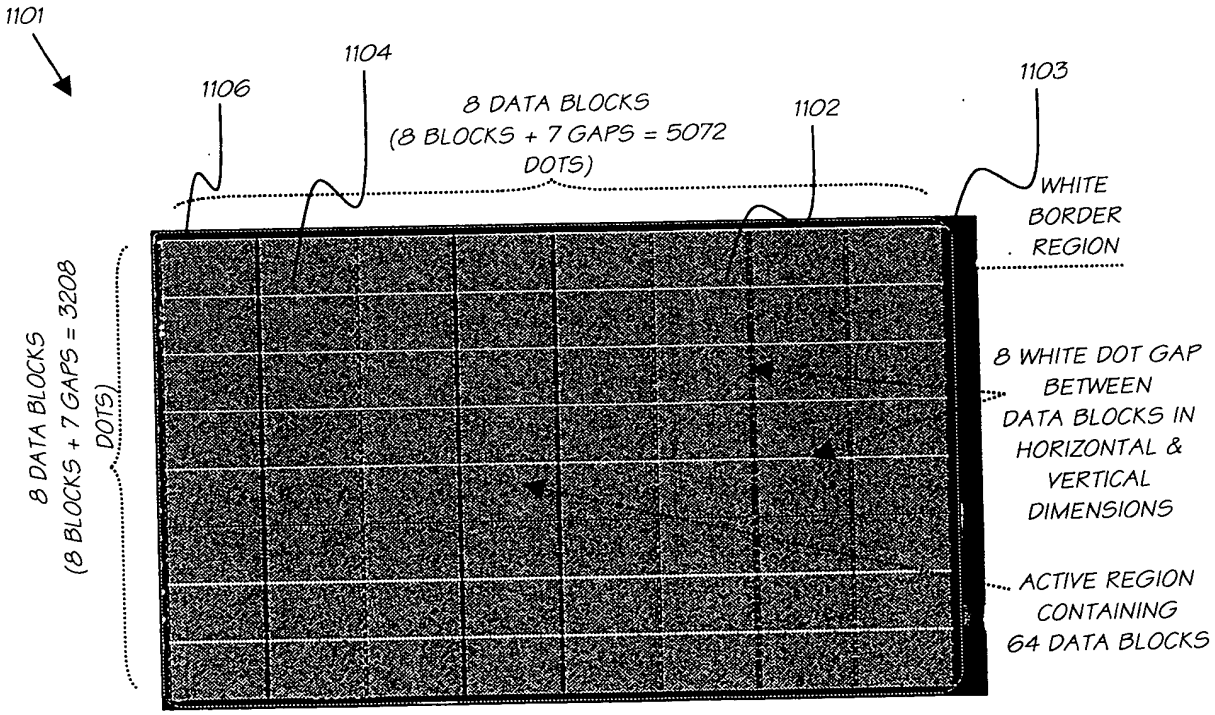
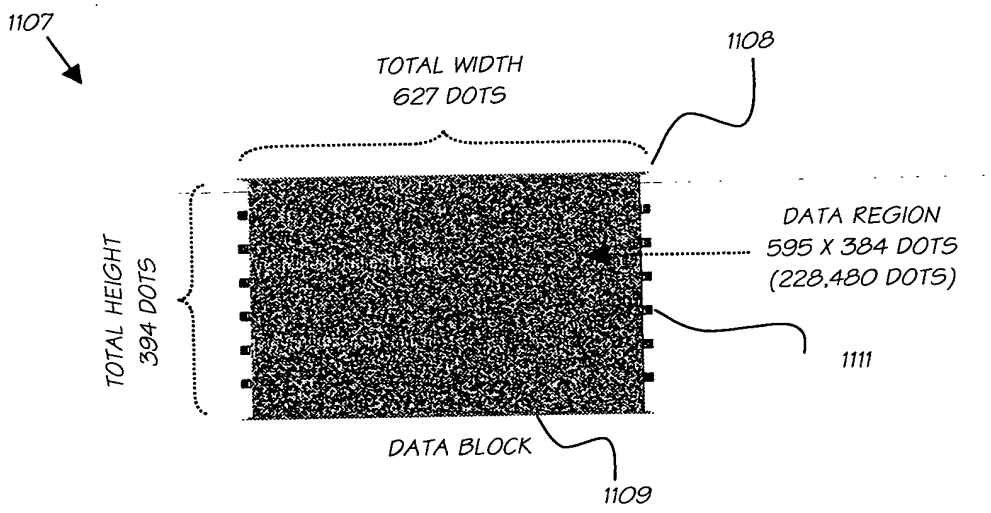
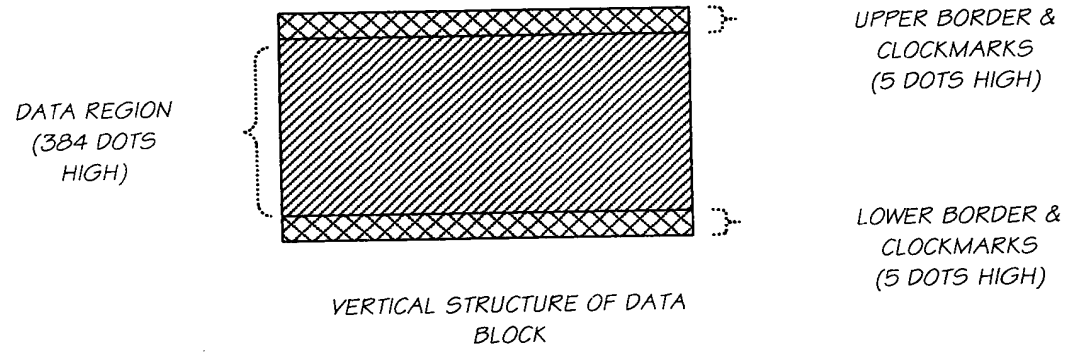

1101
1104
1106
8 DATA BLOCKS (8 BLOCKS + 7 GAPS = 5072 DOTS)
1102
1103
WHITE BORDER REGION
8 DATA BLOCKS (8 BLOCKS + 7 GAPS = 3208 DOTS)
8 WHITE DOT GAP BETWEEN DATA BLOCKS IN HORIZONTAL & VERTICAL DIMENSIONS
ACTIVE REGION CONTAINING 64 DATA BLOCKS

FIG. 48


1107
TOTAL WIDTH
627 DOTS
1108
TOTAL HEIGHT
394 DOTS
DATA REGION
595 X 384 DOTS
(228,480 DOTS)
1111
DATA BLOCK
1109


DATA REGION
(384 DOTS
HIGH)
UPPER BORDER &
CLOCKMARKS
(5 DOTS HIGH)
LOWER BORDER &
CLOCKMARKS
(5 DOTS HIGH)
VERTICAL STRUCTURE OF DATA
BLOCK

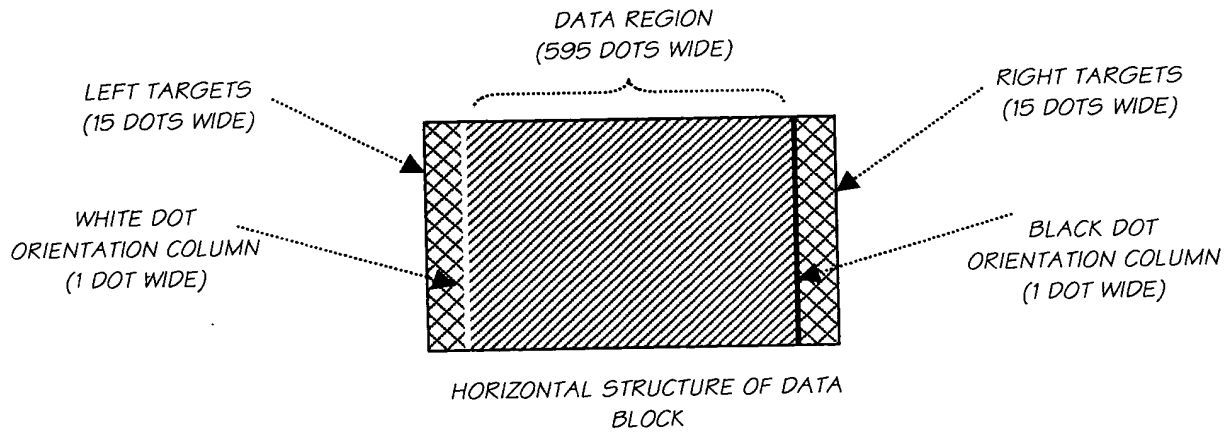

DATA REGION
(595 DOTS WIDE)
LEFT TARGETS
(15 DOTS WIDE)
RIGHT TARGETS
(15 DOTS WIDE)
WHITE DOT
ORIENTATION COLUMN
(1 DOT WIDE)
BLACK DOT
ORIENTATION COLUMN
(1 DOT WIDE)
HORIZONTAL STRUCTURE OF DATA
BLOCK

FIG. 49

FIG. 51

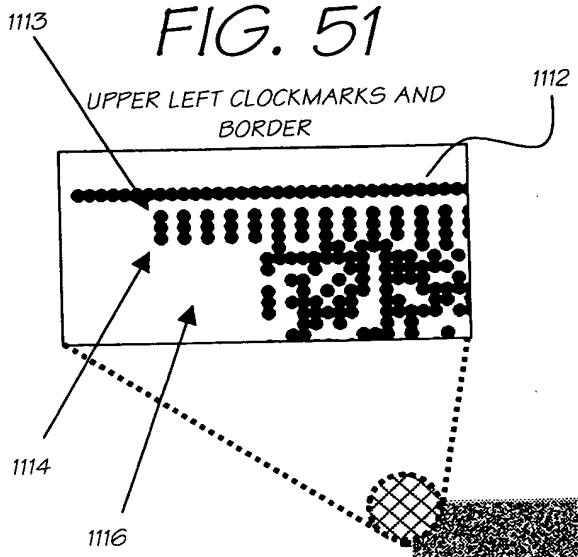
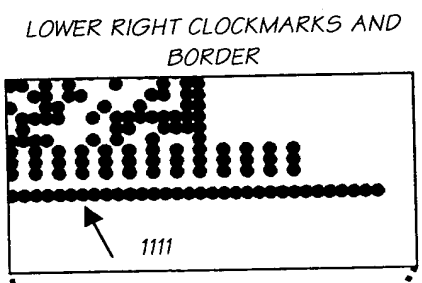
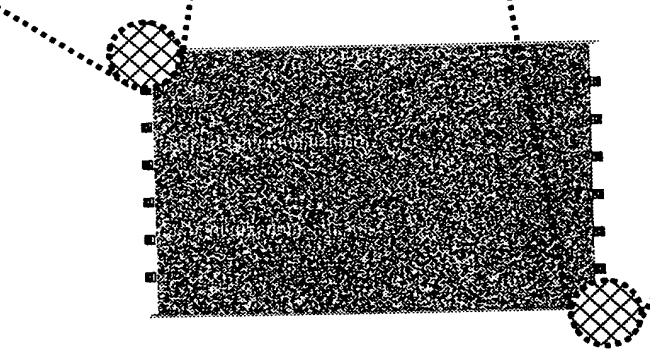


FIG. 52





1107

FIG. 50

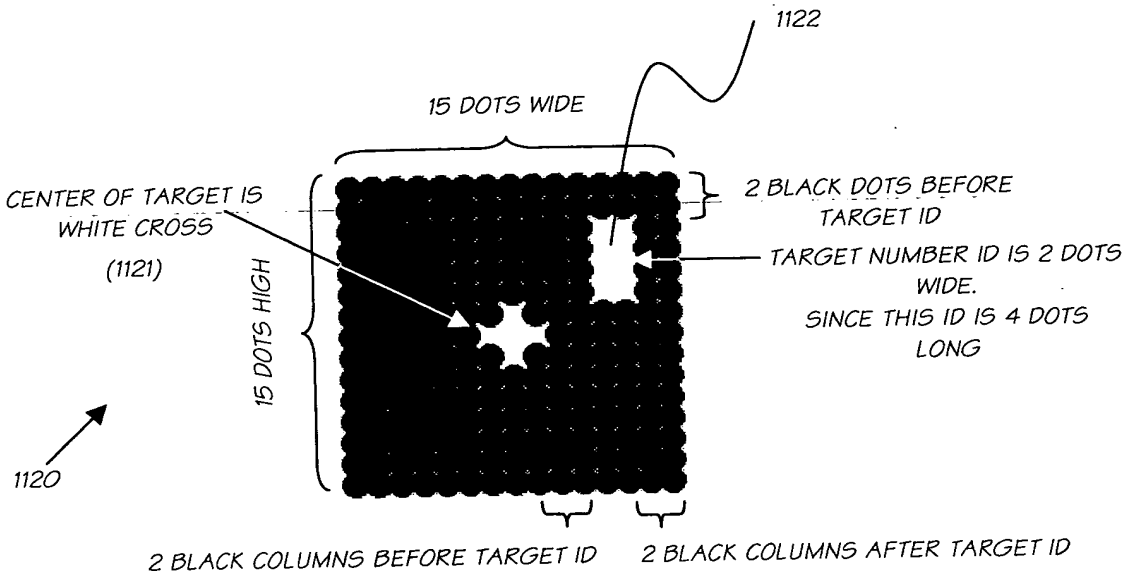

1122
15 DOTS WIDE
CENTER OF TARGET IS WHITE CROSS (1121)
2 BLACK DOTS BEFORE TARGET ID
TARGET NUMBER ID IS 2 DOTS WIDE. SINCE THIS ID IS 4 DOTS LONG
15 DOTS HIGH
1120
2 BLACK COLUMNS BEFORE TARGET ID
2 BLACK COLUMNS AFTER TARGET ID

FIG. 53

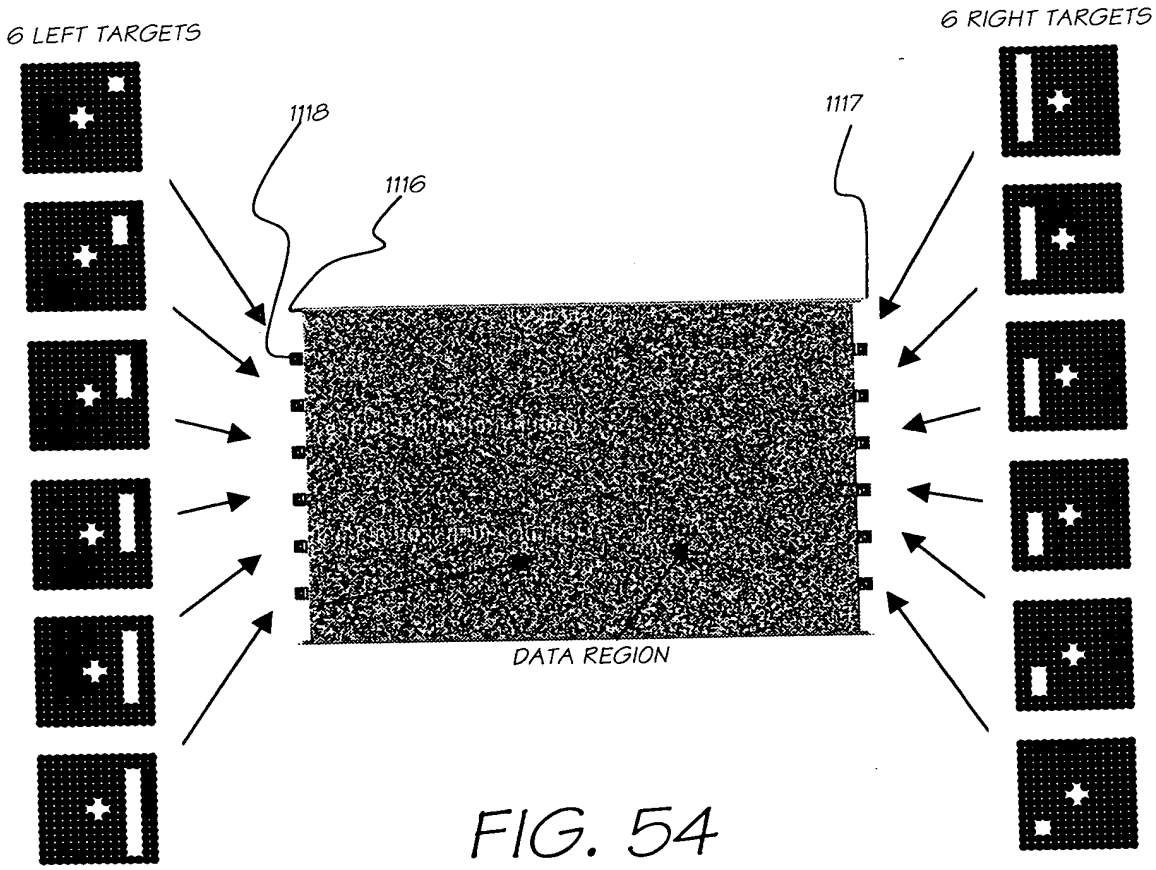

6 LEFT TARGETS
6 RIGHT TARGETS
1118
1116
1117
DATA REGION

FIG. 54

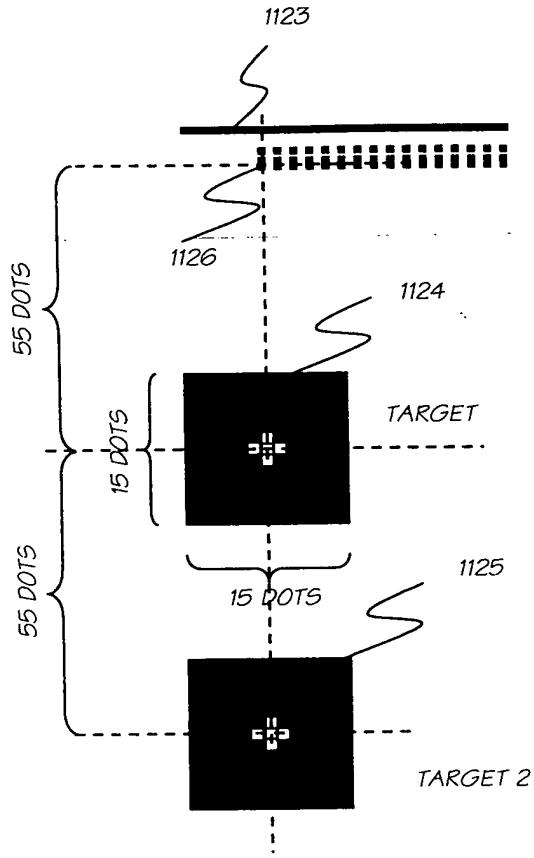

1123
1126
55 DOTS
1124
TARGET
15 DOTS
55 DOTS
1125
15 DOTS
TARGET 2

FIG. 55

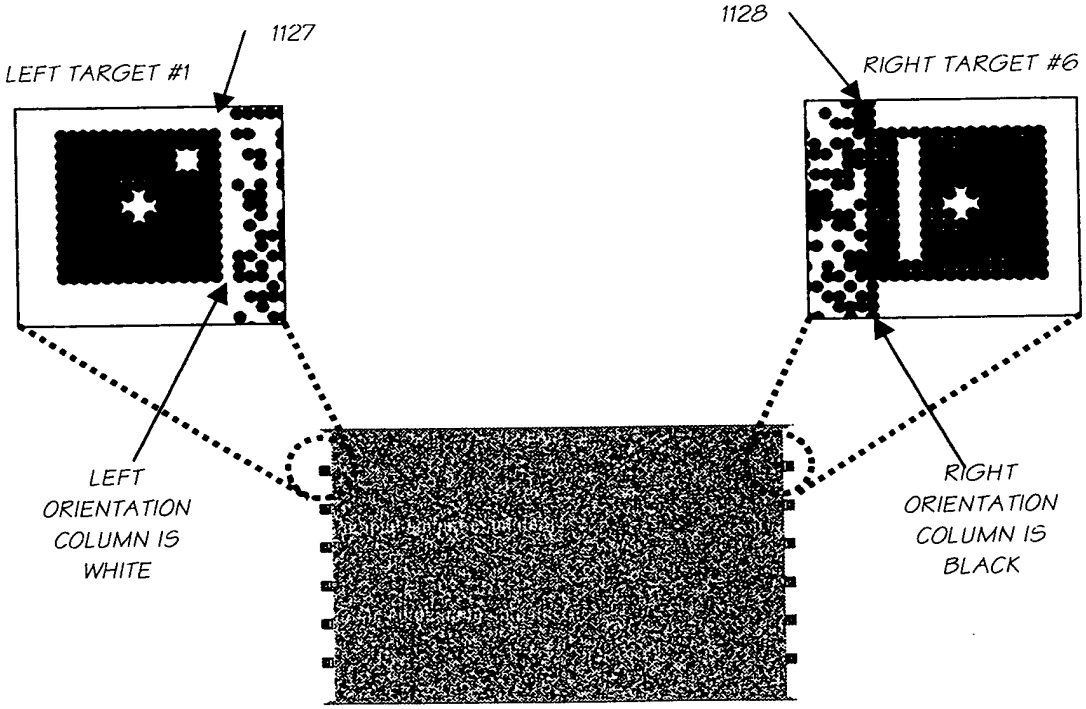

1127
LEFT TARGET #1
1128
RIGHT TARGET #6
LEFT ORIENTATION COLUMN IS WHITE
RIGHT ORIENTATION COLUMN IS BLACK

FIG. 56

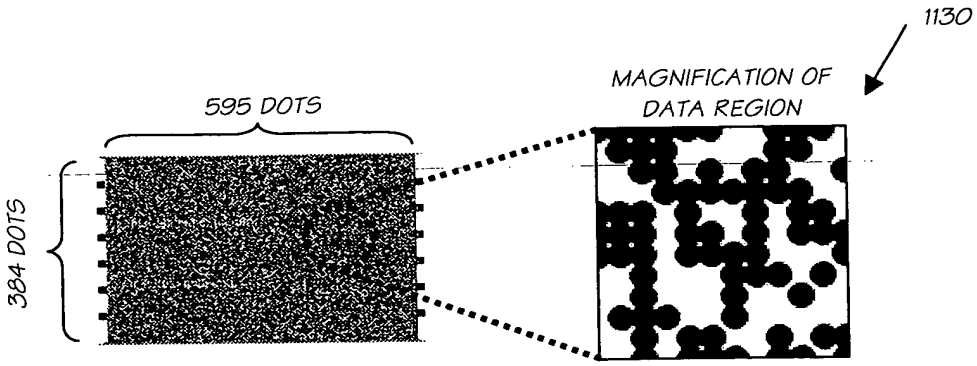


FIG. 57

00:	4F	00	3D	4F	00	3D	4F	00	3D	4F	00	3D	32 COPIES OF THE 3 BYTE CONTROL INFORMATION
0C:	4F	00	3D	4F	00	3D	4F	00	3D	4F	00	3D	
18:	4F	00	3D	4F	00	3D	4F	00	3D	4F	00	3D	
24:	4F	00	3D	4F	00	3D	4F	00	3D	4F	00	3D	
30:	4F	00	3D	4F	00	3D	4F	00	3D	4F	00	3D	
3C:	4F	00	3D	4F	00	3D	4F	00	3D	4F	00	3D	
48:	4F	00	3D	4F	00	3D	4F	00	3D	4F	00	3D	
54:	4F	00	3D	4F	00	3D	4F	00	3D	4F	00	3D	
60:	00	00	00	00	00	00	00	00	00	00	00	00	RESERVED BYTES ARE 0
6C:	00	00	00	00	00	00	00	00	00	00	00	00	
78:	00	00	00	00	00	00	00	00	00	00	00		

FIG. 59

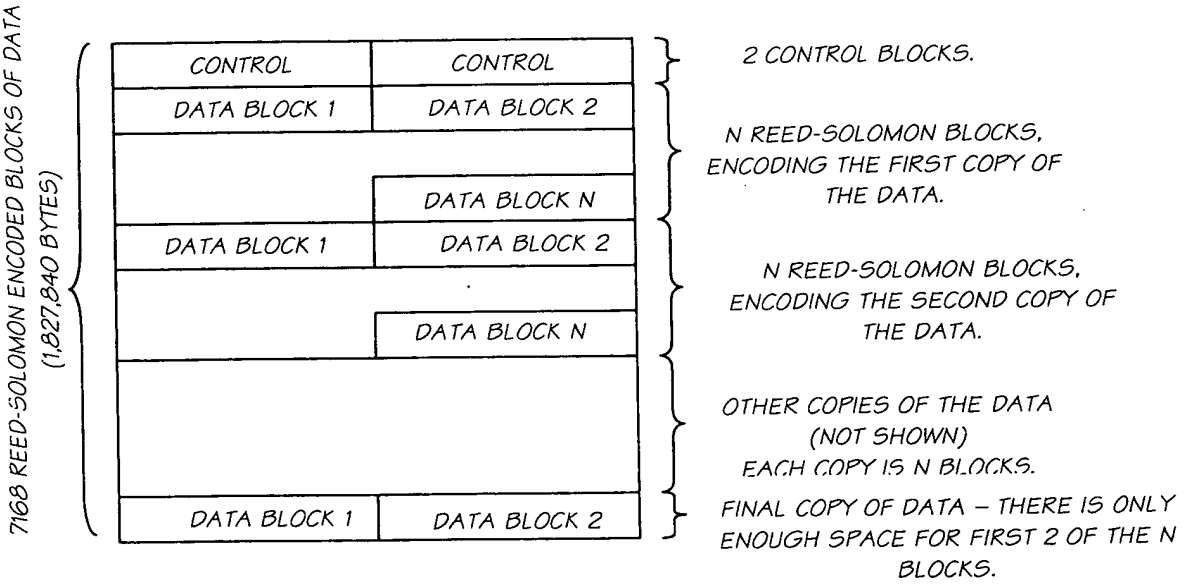


FIG. 58

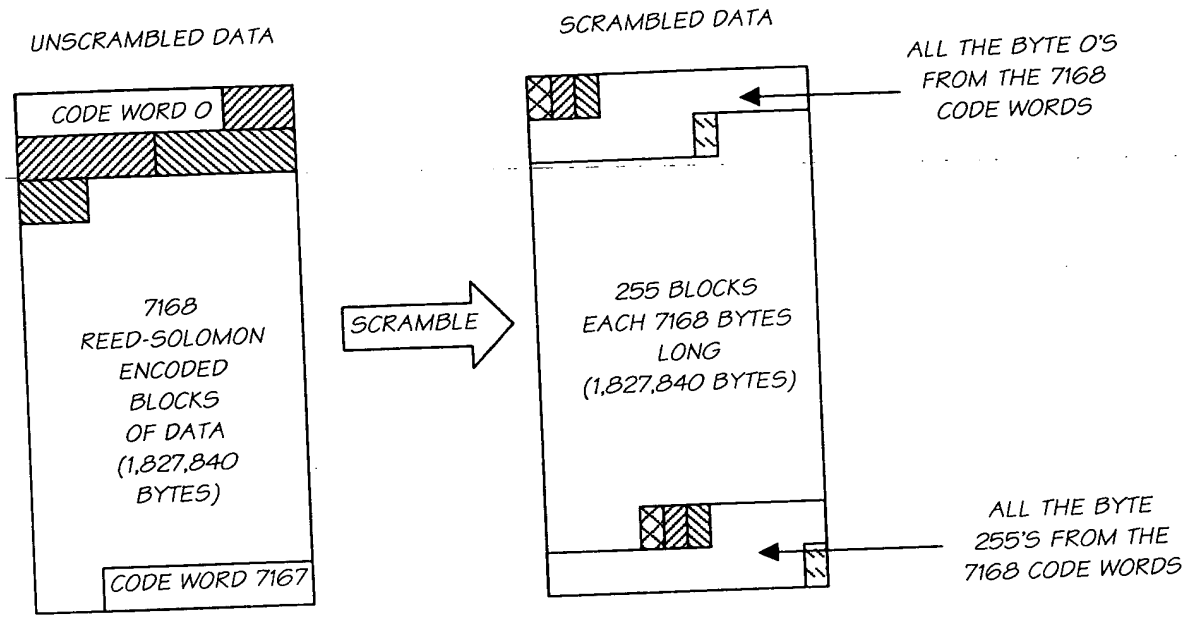


FIG. 60

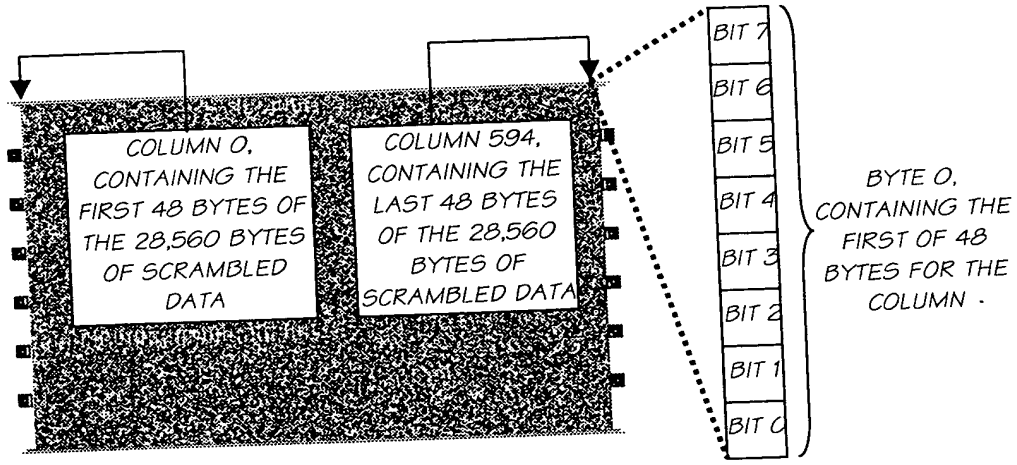


FIG. 61

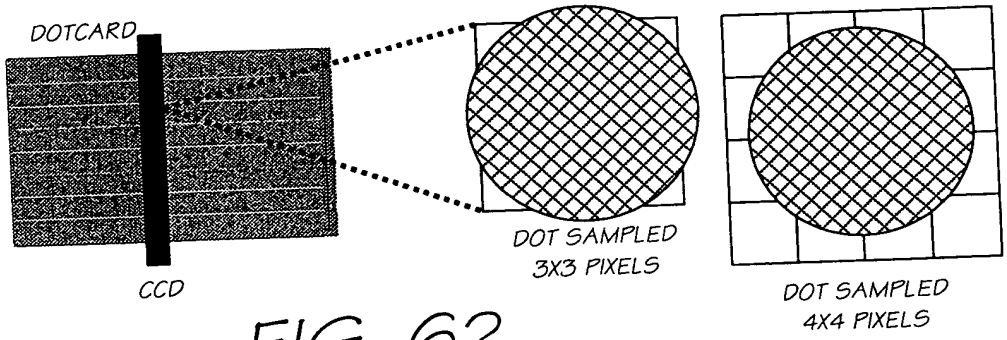


FIG. 62

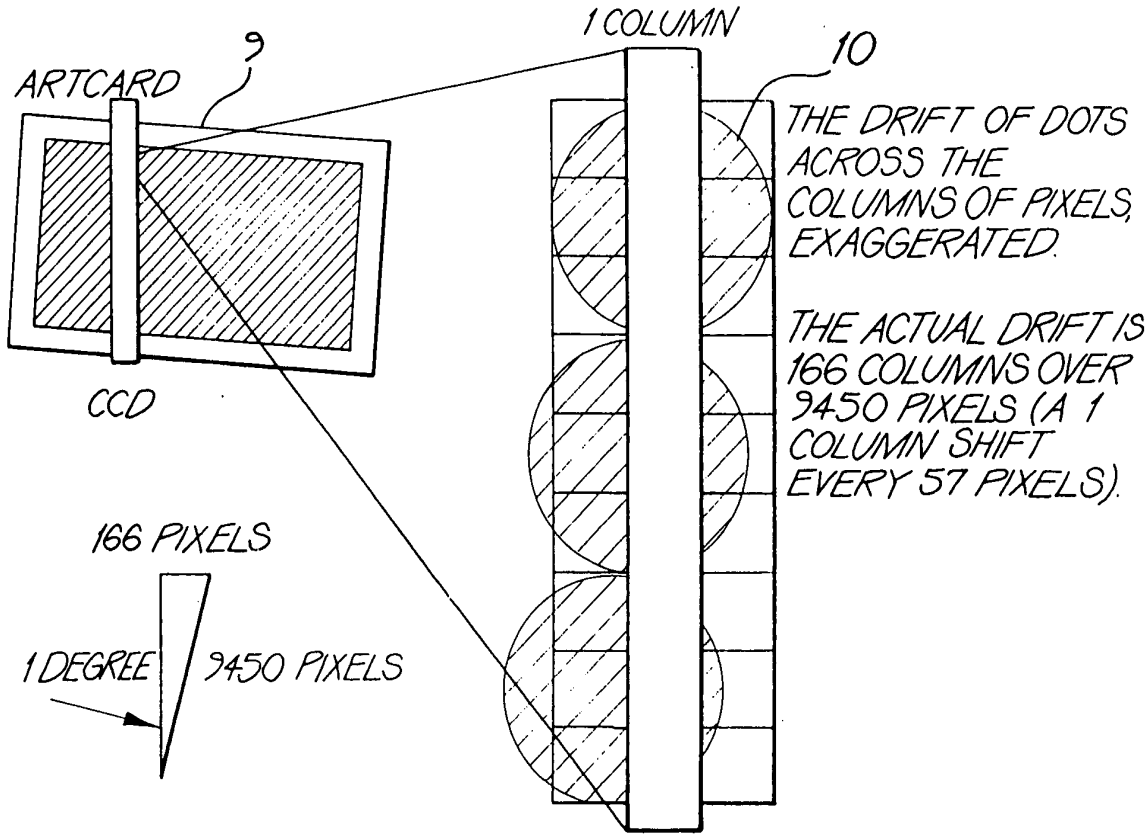

1 COLUMN
9
ARTCARD
10
THE DRIFT OF DOTS ACROSS THE COLUMNS OF PIXELS, EXAGGERATED.
THE ACTUAL DRIFT IS 166 COLUMNS OVER 9450 PIXELS (A 1 COLUMN SHIFT EVERY 57 PIXELS).
CCD
166 PIXELS
1 DEGREE
9450 PIXELS

FIG. 63

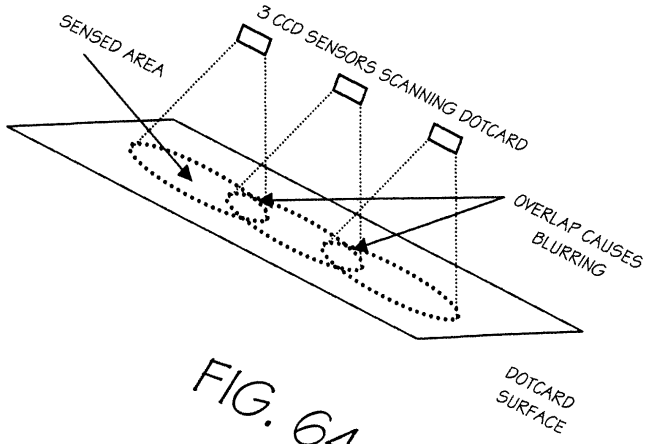

SENSED AREA
3 CCD SENSORS SCANNING DOTCARD
OVERLAP CAUSES BLURRING
DOTCARD SURFACE

FIG. 64

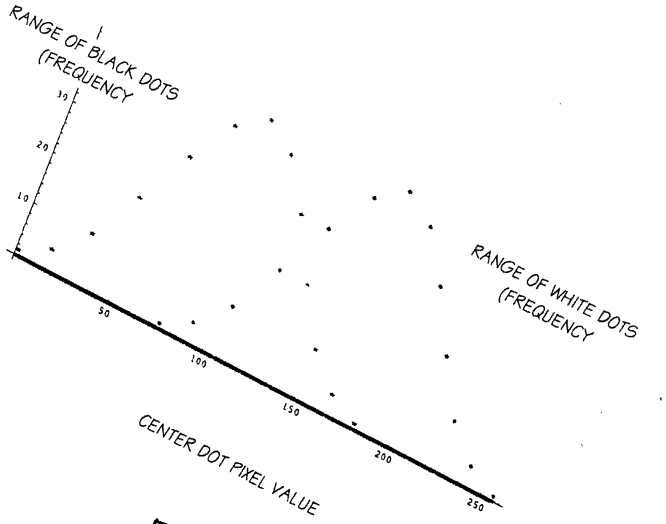
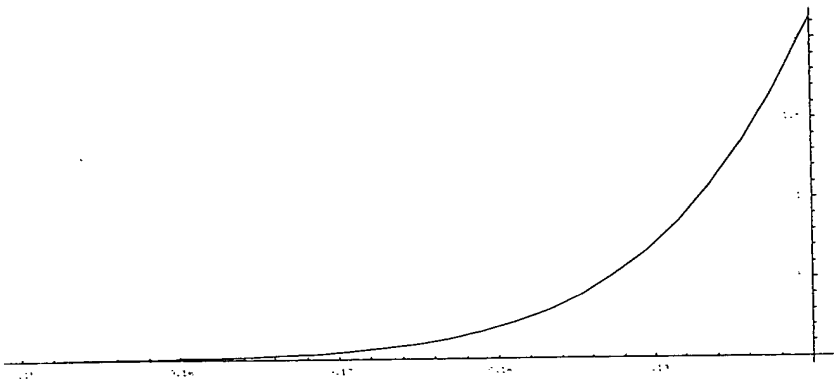
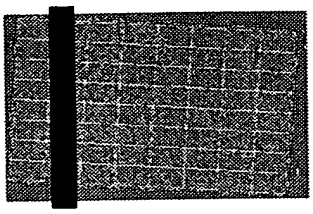

RANGE OF BLACK DOTS (FREQUENCY
30
20
10
50
100
150
200
250
RANGE OF WHITE DOTS (FREQUENCY
CENTER DOT PIXEL VALUE

FIG. 65



PROBABILITY OF A SYMBOL BEING IN ERROR DURING A READ

FIG. 66



READ LINEAR CCD

DOTCARD PIXEL DATA

EXTRACT BIT IMAGE

ENCODED, SCRAMBLED, POTENTIALLY

ROTATE AND UNSCRAMBLE

ENCODED, UNSCRAMBLED DATA

REED-SOLOMON DECODE

RAW DATA

APPROXIMATE DATA SIZES FOR 1600 DPI DOTCARD

86MM + 1MM IN HORIZONTAL DIMENSION FOR 1° ROTATION = 87MM

87MM = 16,252 SCANLINES

16,440 SCANLINES @ 11,000 PIXELS PER SCANLINE = 180,840,000 PIXELS

180,840,000 PIXELS @ 1 BYTE PER PIXEL = 180,840,000 BYTES = 172.5 MB

64 DATA BLOCKS, EACH CONTAINING 597 COLUMNS (595 DATA REGION COLUMNS AND 2 ORIENTATION COLUMNS), @ 48 BYTES PER COLUMN = 28,656 BYTES PER DATA BLOCK FOR A TOTAL OF 1,833,984 BYTES.

64 DATA BLOCKS, EACH CONTAINING 112 ENCODED REED SOLOMON BLOCKS, @ 255 BYTES PER REED SOLOMON BLOCK FOR A TOTAL OF 1,827,840 BYTES.

DECODED DATA, WITH A MAXIMUM SIZE OF 910,082 BYTES. (64 X 112 X 127 – (2 CONTROL BLOCKS @ 127 BYTES))

FIG. 67

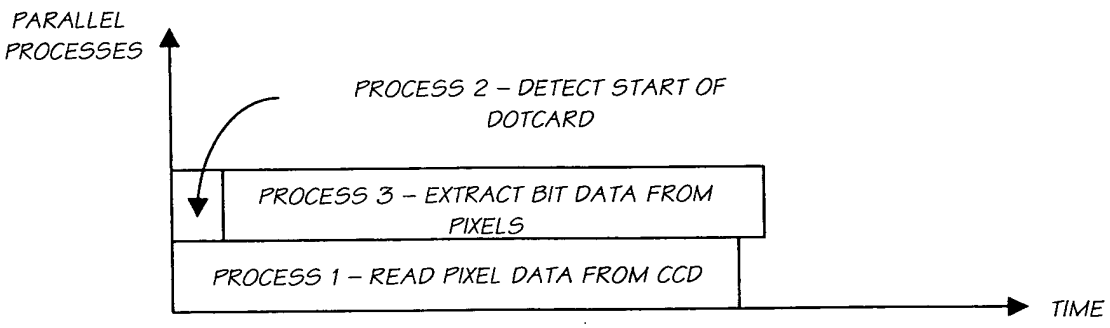

PARALLEL PROCESSES
PROCESS 2 – DETECT START OF DOTCARD
PROCESS 3 – EXTRACT BIT DATA FROM PIXELS
PROCESS 1 – READ PIXEL DATA FROM CCD
TIME

FIG. 68

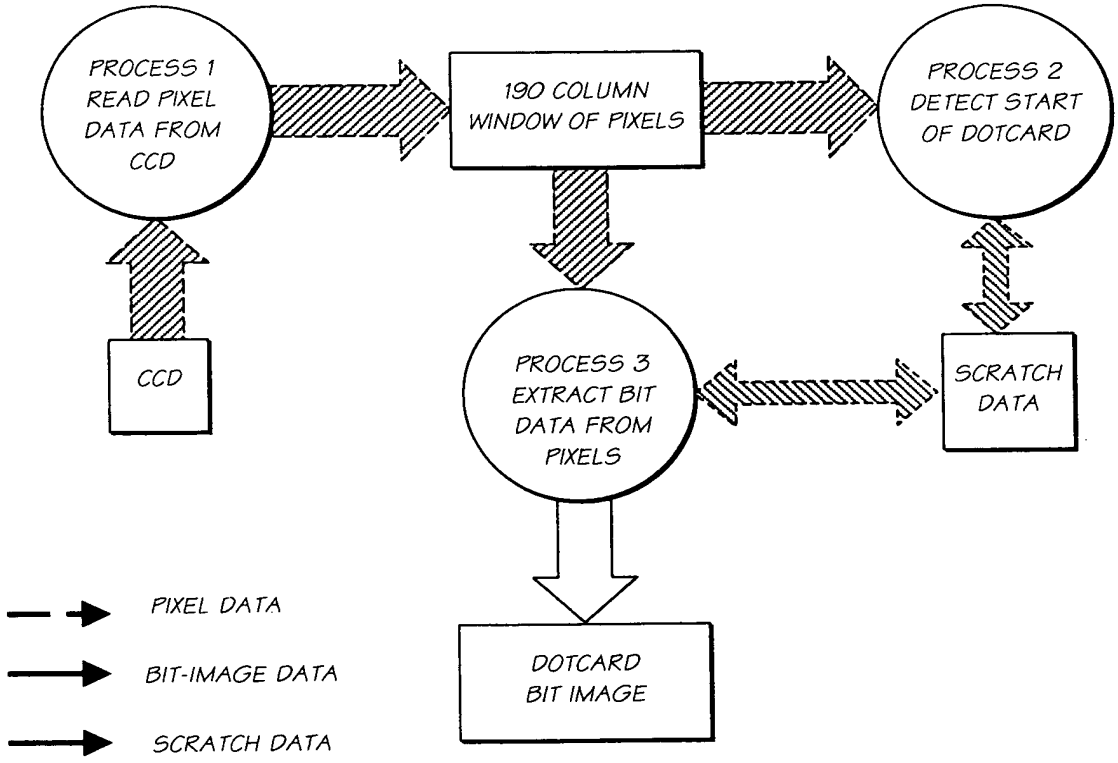

PROCESS 1 READ PIXEL DATA FROM CCD
190 COLUMN WINDOW OF PIXELS
PROCESS 2 DETECT START OF DOTCARD
CCD
PROCESS 3 EXTRACT BIT DATA FROM PIXELS
SCRATCH DATA
DOTCARD BIT IMAGE
PIXEL DATA
BIT-IMAGE DATA
SCRATCH DATA

FIG. 69

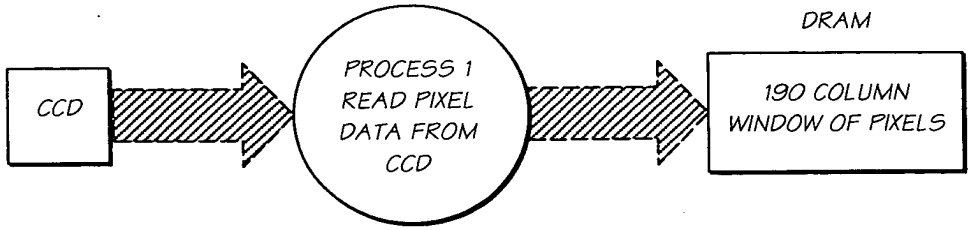


FIG. 70

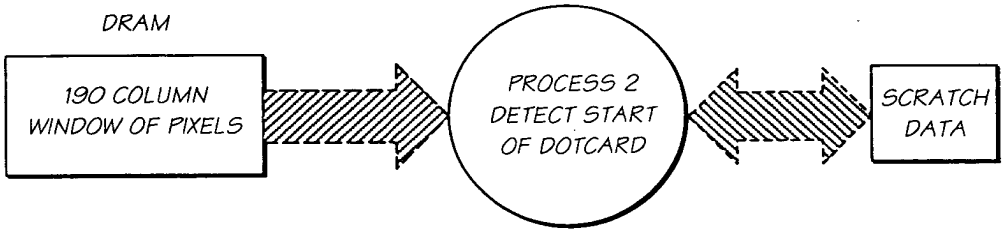


FIG. 71

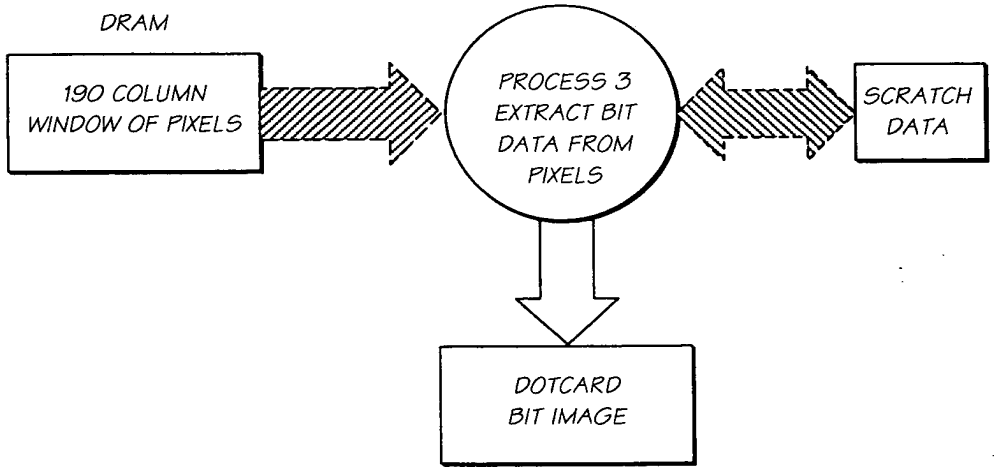


FIG. 72

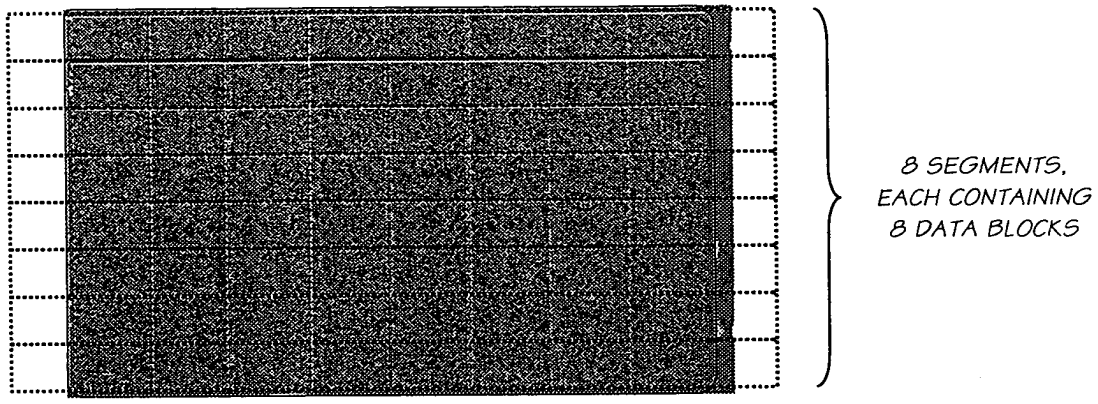

8 SEGMENTS,
EACH CONTAINING
8 DATA BLOCKS

FIG. 73

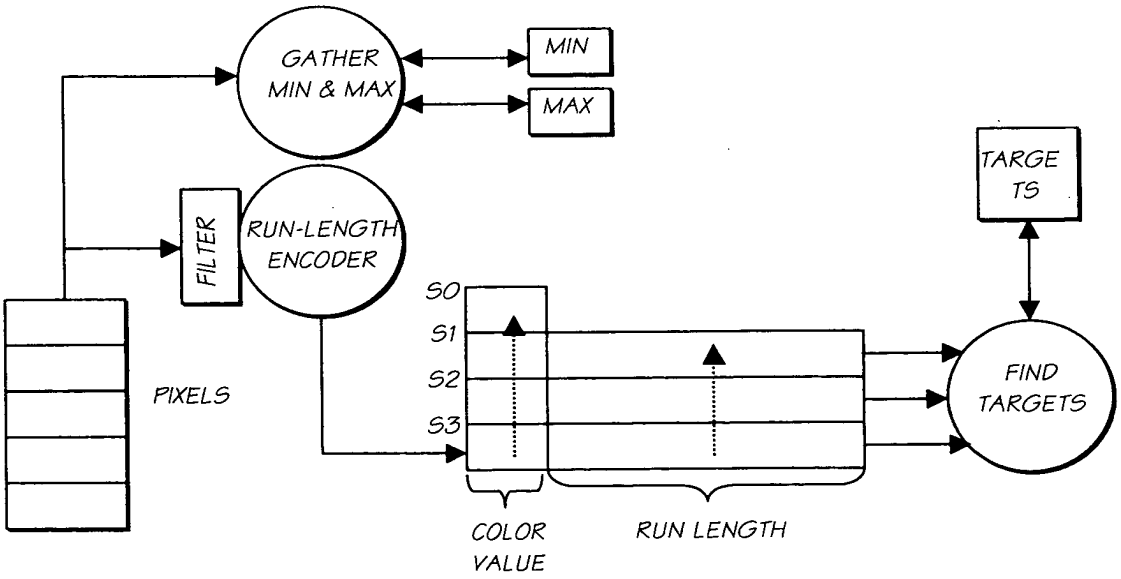

GATHER
MIN & MAX
MIN
MAX
TARGE
TS
FILTER
RUN-LENGTH
ENCODER
S0
S1
S2
S3
PIXELS
FIND
TARGETS
COLOR
VALUE
RUN LENGTH

FIG. 74

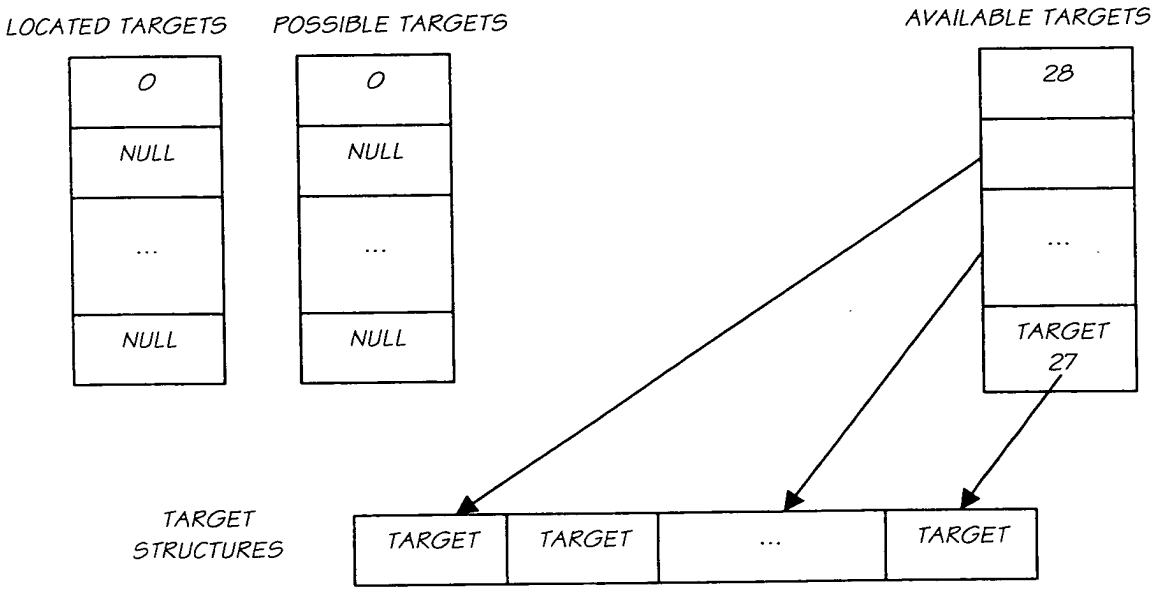
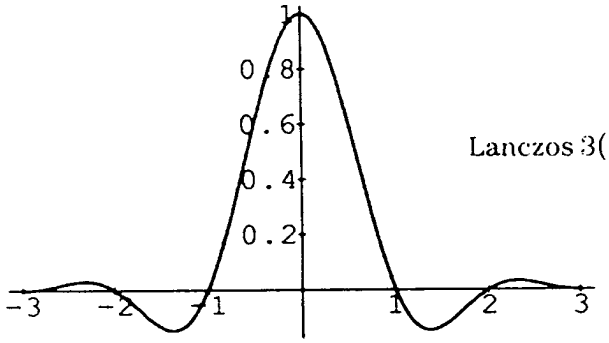


FIG. 75



$$\text{Lanczos}\,3(x) = \begin{cases} \dfrac{\sin(\pi x)}{\pi x}\,\dfrac{\sin\left(\pi\dfrac{x}{3}\right)}{\pi\dfrac{x}{3}}, & |x| < 3 \\ 0, & |x| \geq 3 \end{cases}$$

FIG. 76

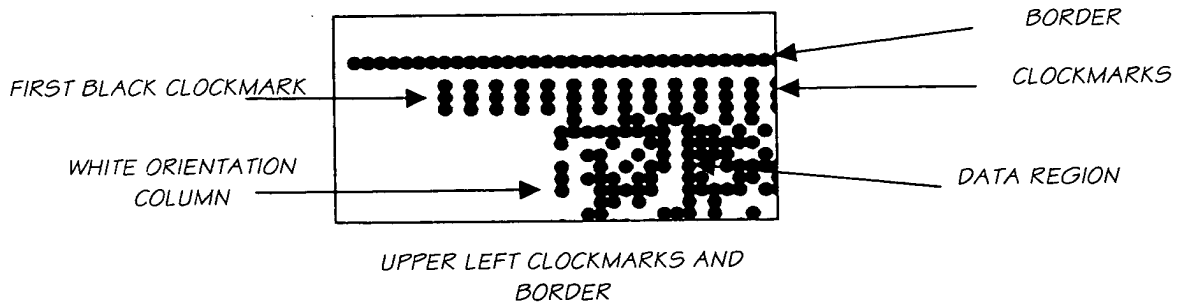


FIG. 77

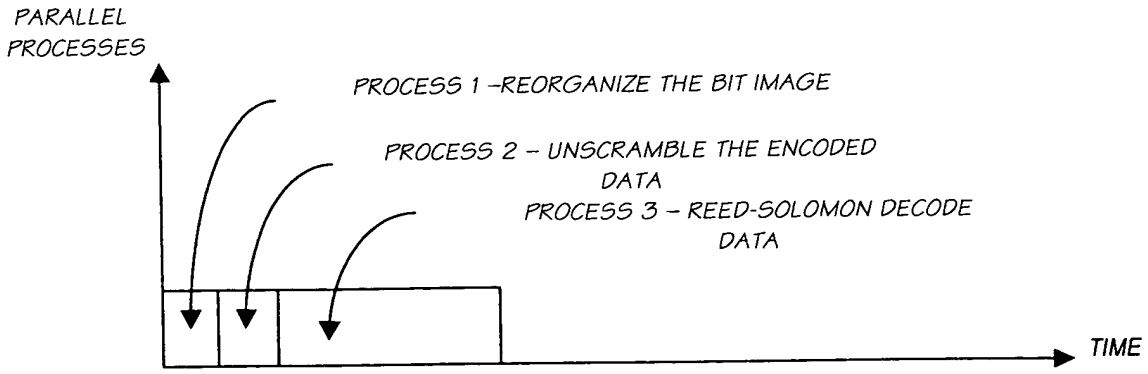

PARALLEL PROCESSES
PROCESS 1 –REORGANIZE THE BIT IMAGE
PROCESS 2 – UNSCRAMBLE THE ENCODED DATA
PROCESS 3 – REED-SOLOMON DECODE DATA
TIME

FIG. 78

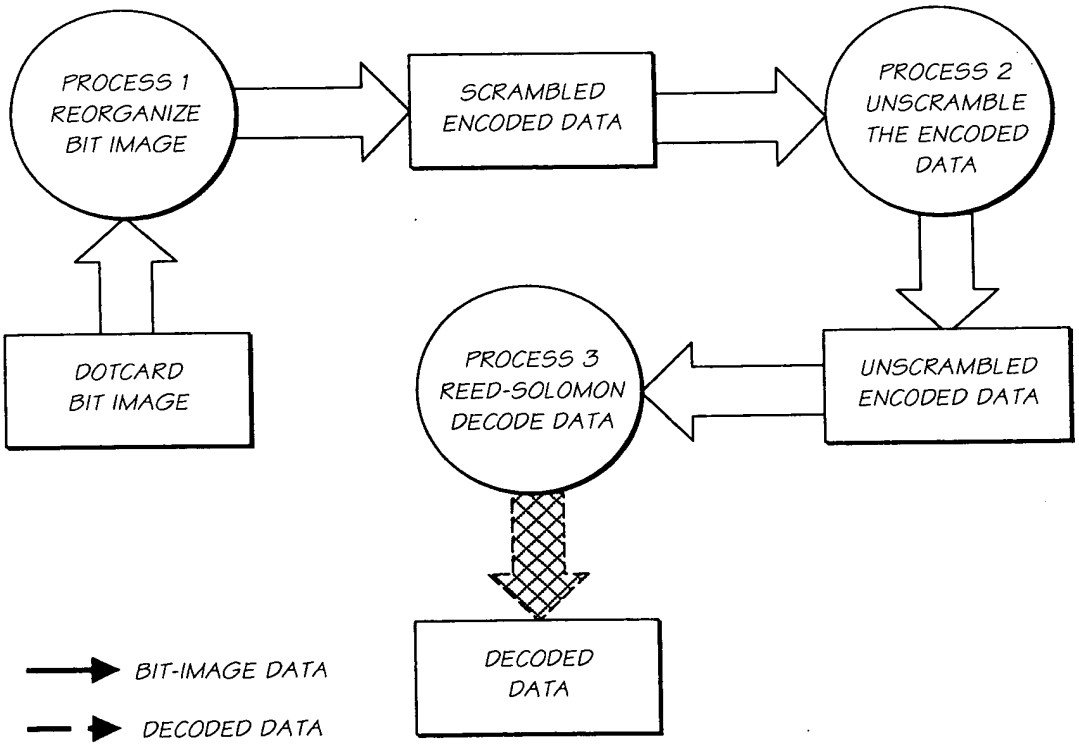

PROCESS 1 REORGANIZE BIT IMAGE
SCRAMBLED ENCODED DATA
PROCESS 2 UNSCRAMBLE THE ENCODED DATA
DOTCARD BIT IMAGE
PROCESS 3 REED-SOLOMON DECODE DATA
UNSCRAMBLED ENCODED DATA
DECODED DATA
BIT-IMAGE DATA
DECODED DATA

FIG. 79

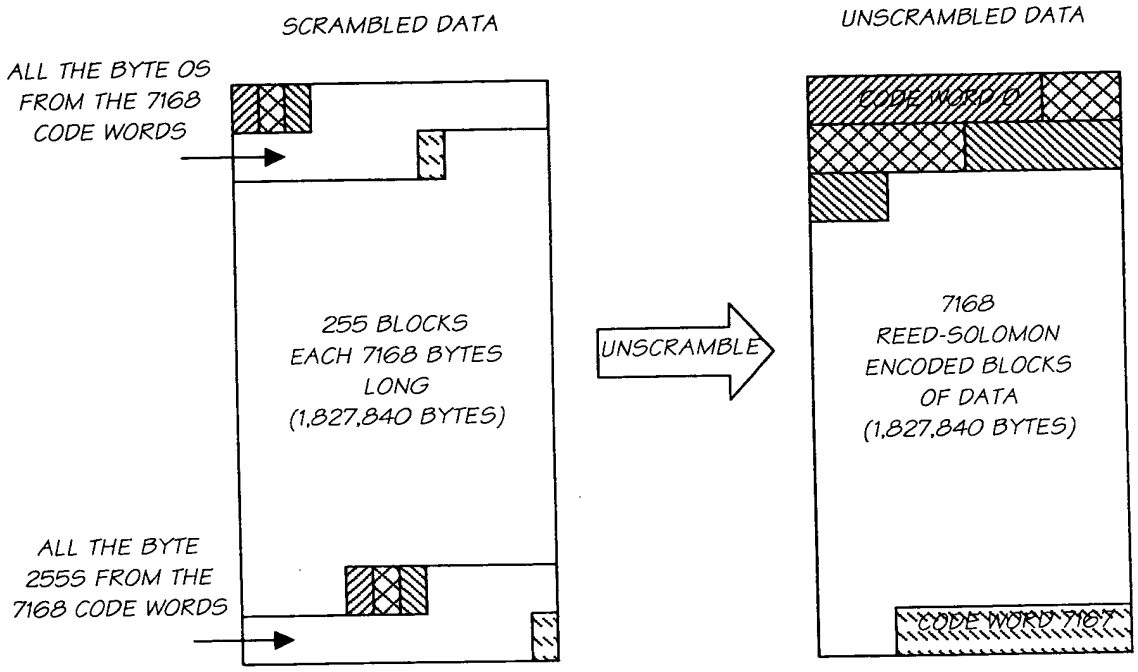

SCRAMBLED DATA
UNSCRAMBLED DATA
ALL THE BYTE OS FROM THE 7168 CODE WORDS
255 BLOCKS EACH 7168 BYTES LONG (1,827,840 BYTES)
ALL THE BYTE 2555 FROM THE 7168 CODE WORDS
UNSCRAMBLE
CODE WORD 0
7168 REED-SOLOMON ENCODED BLOCKS OF DATA (1,827,840 BYTES)
CODE WORD 7167

FIG. 80

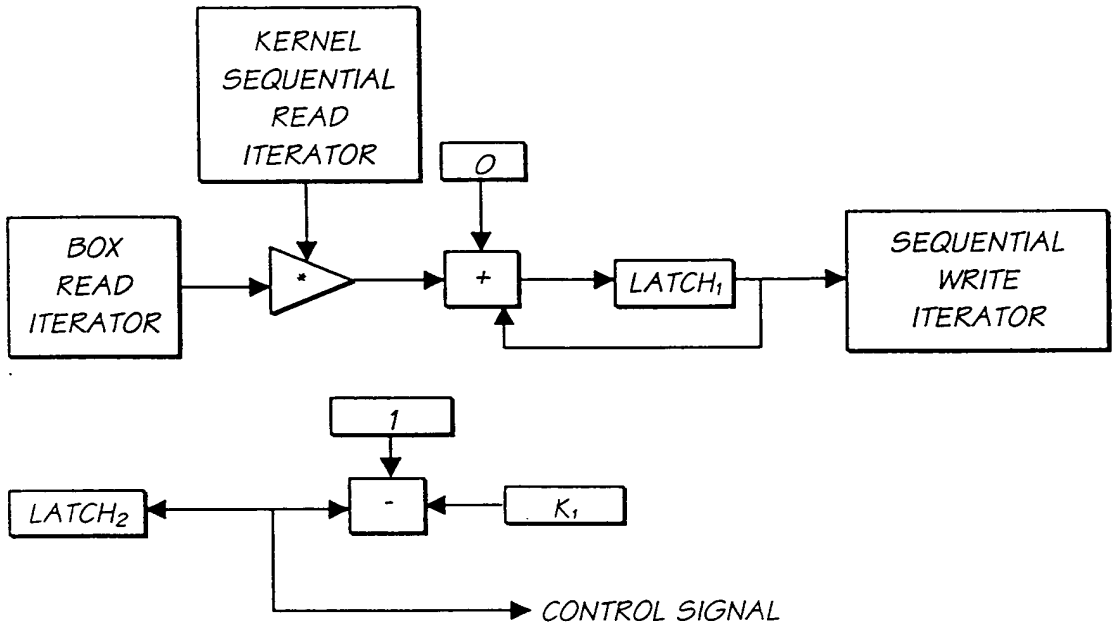

KERNEL SEQUENTIAL READ ITERATOR
0
BOX READ ITERATOR
*
+
LATCH1
SEQUENTIAL WRITE ITERATOR
1
LATCH2
-
K1
CONTROL SIGNAL

FIG. 81

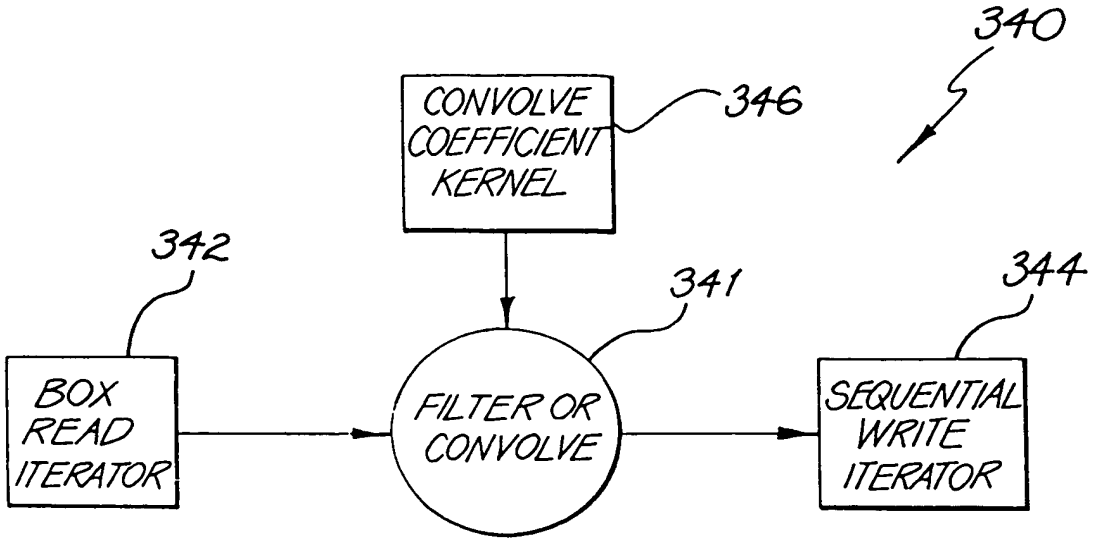

340
CONVOLVE COEFFICIENT KERNEL
346
342
341
344
BOX READ ITERATOR
FILTER OR CONVOLVE
SEQUENTIAL WRITE ITERATOR

FIG. 82

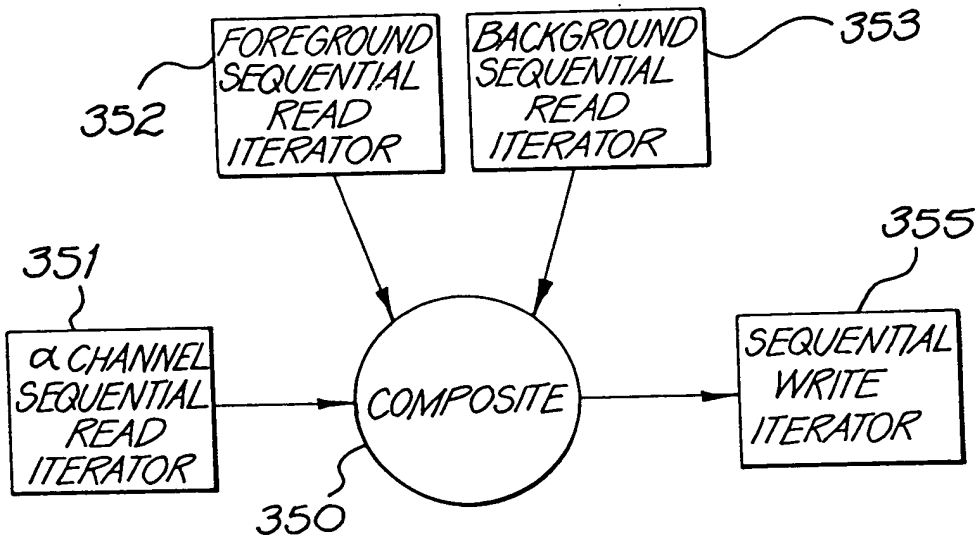

FOREGROUND SEQUENTIAL READ ITERATOR
352
BACKGROUND SEQUENTIAL READ ITERATOR
353
355
351
α CHANNEL SEQUENTIAL READ ITERATOR
COMPOSITE
SEQUENTIAL WRITE ITERATOR
350

FIG. 83

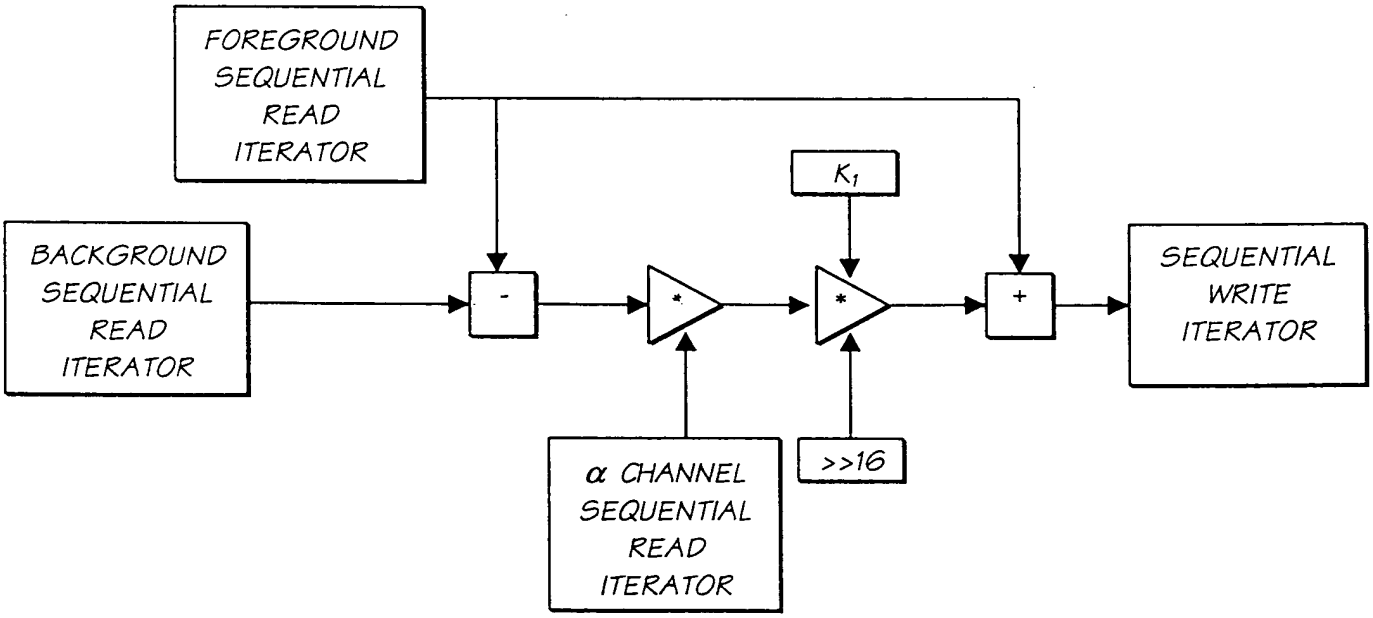

FOREGROUND SEQUENTIAL READ ITERATOR
K_1
BACKGROUND SEQUENTIAL READ ITERATOR
-
*
*
+
SEQUENTIAL WRITE ITERATOR
α CHANNEL SEQUENTIAL READ ITERATOR
>>16

FIG. 84

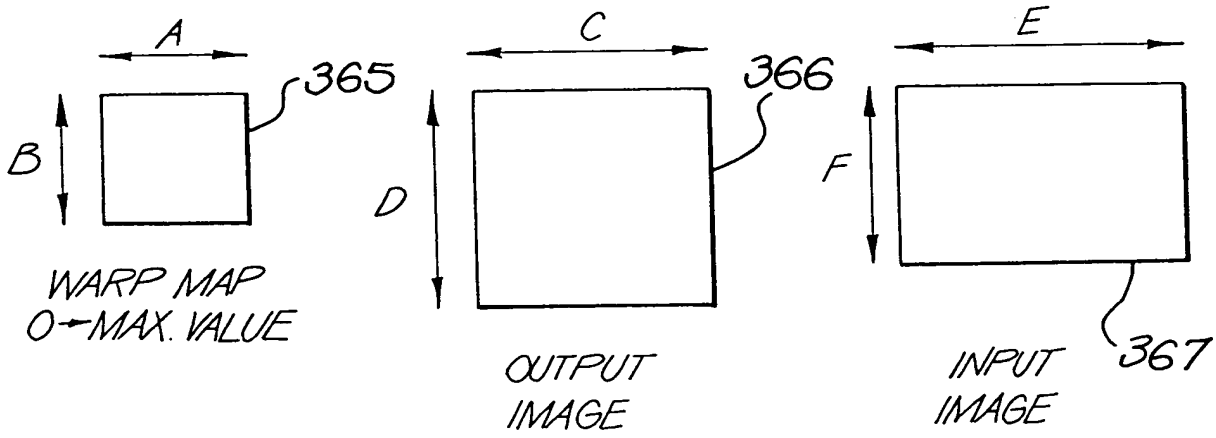

A
365
B
WARP MAP
0→MAX. VALUE
C
366
D
OUTPUT
IMAGE
E
F
INPUT
IMAGE
367

FIG. 85

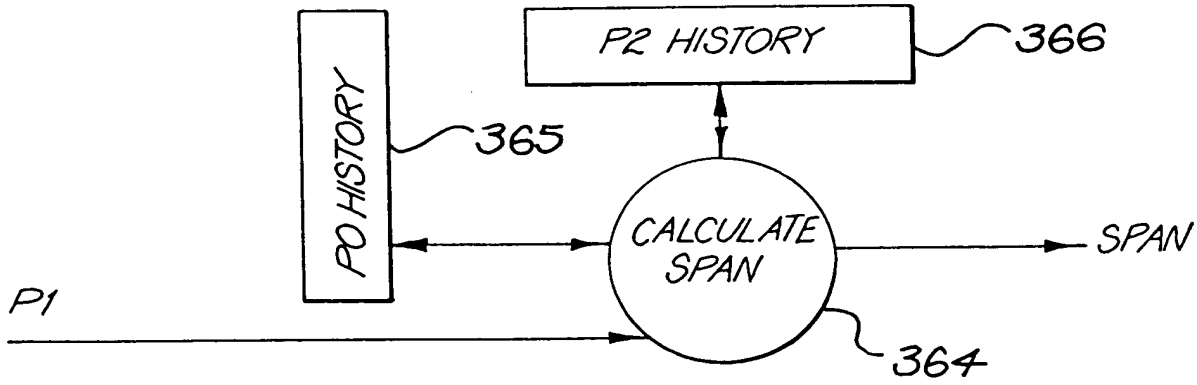

P2 HISTORY
366
P0 HISTORY
365
CALCULATE
SPAN
SPAN
P1
364

FIG. 86

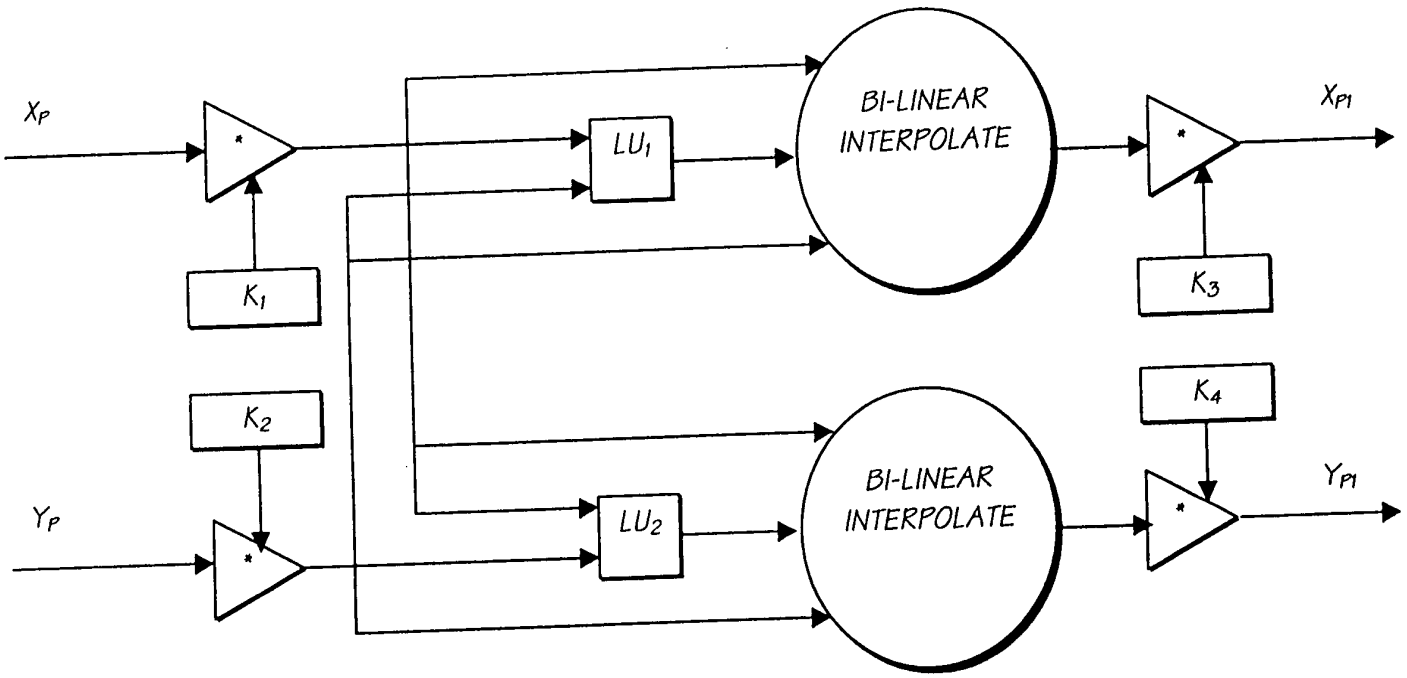

X_P
LU_1
BI-LINEAR INTERPOLATE
X_{P1}
K_1
K_3
K_2
K_4
Y_P
LU_2
BI-LINEAR INTERPOLATE
Y_{P1}

FIG. 87

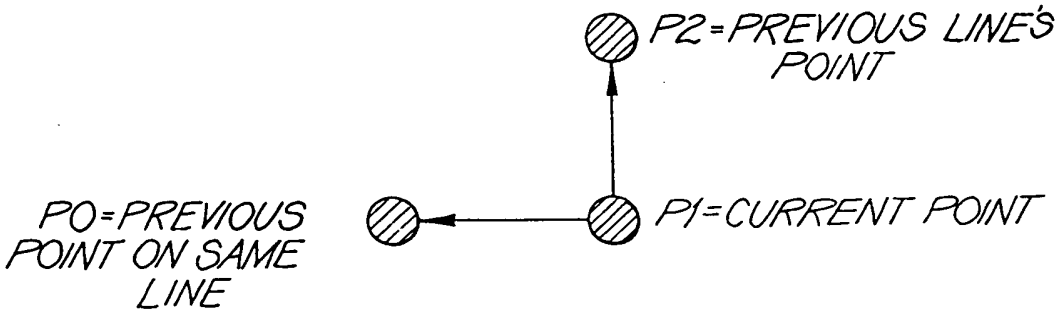

P2=PREVIOUS LINE'S POINT
P0=PREVIOUS POINT ON SAME LINE
P1=CURRENT POINT

FIG. 88

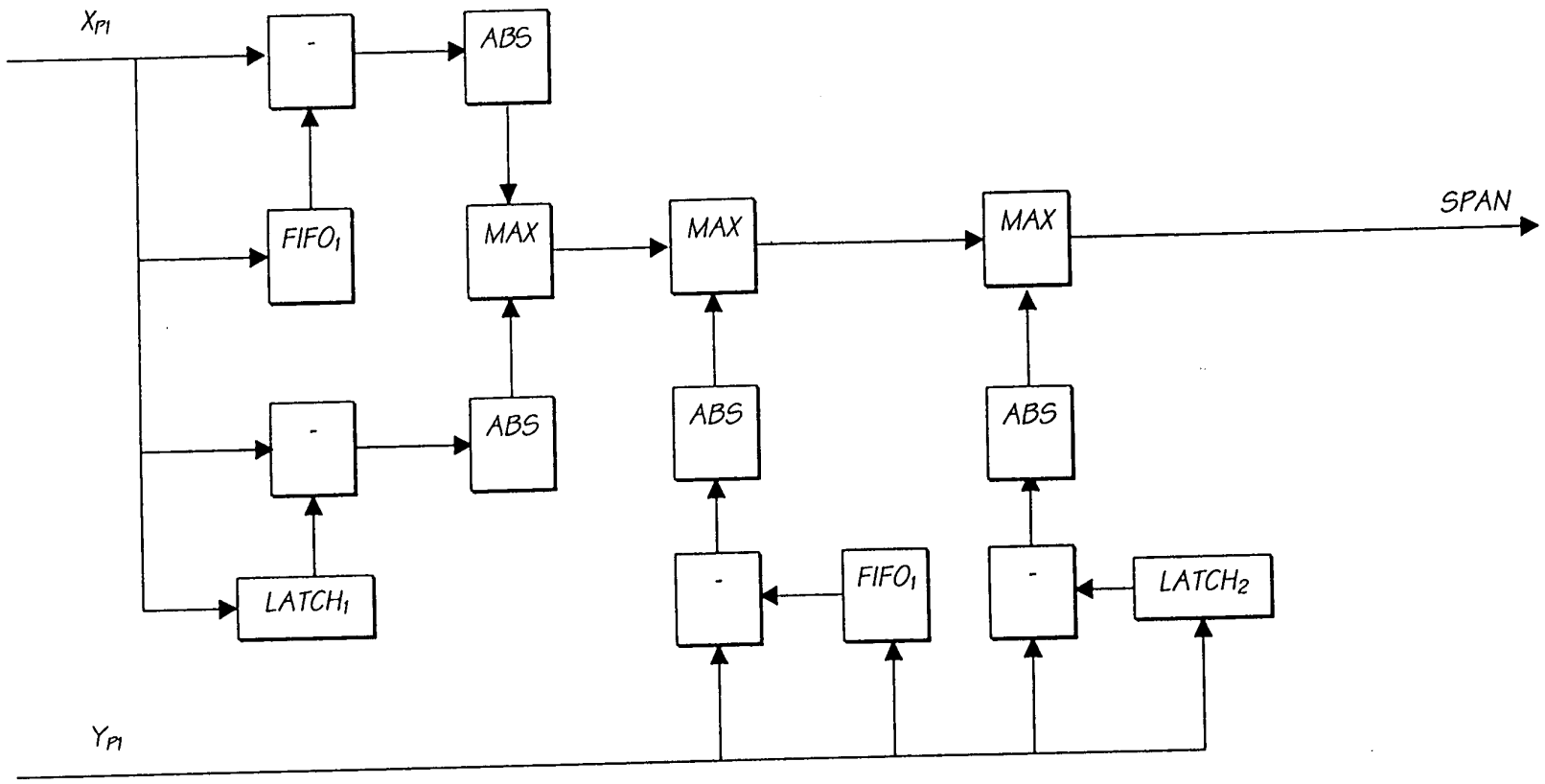

X_{P1}
-
ABS
$FIFO_1$
MAX
MAX
MAX
SPAN
-
ABS
ABS
ABS
$LATCH_1$
-
$FIFO_1$
-
$LATCH_2$
Y_{P1}

FIG. 89

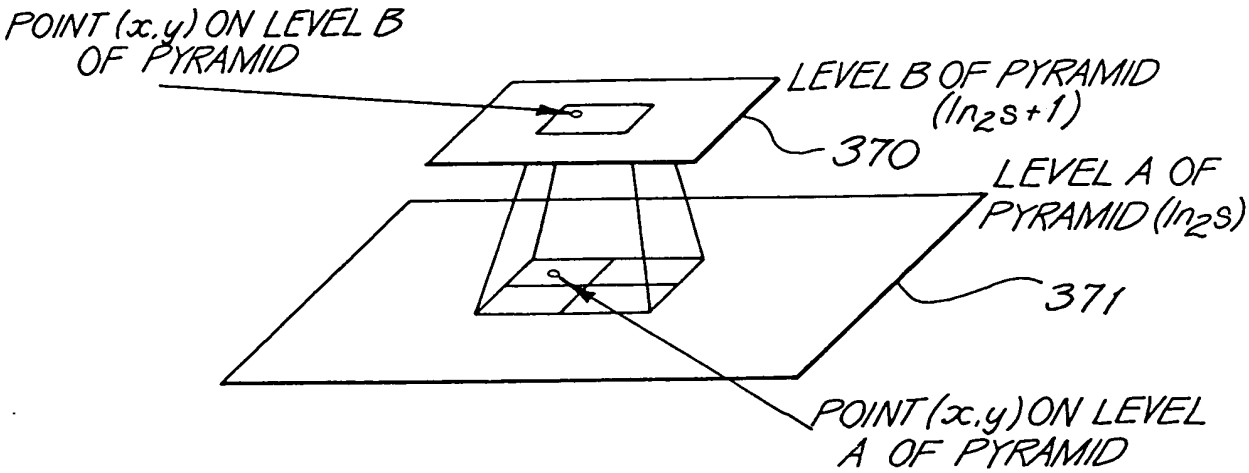

POINT (x,y) ON LEVEL B OF PYRAMID
LEVEL B OF PYRAMID ($ln_2 s+1$)
370
LEVEL A OF PYRAMID ($ln_2 s$)
371
POINT (x,y) ON LEVEL A OF PYRAMID

FIG. 90

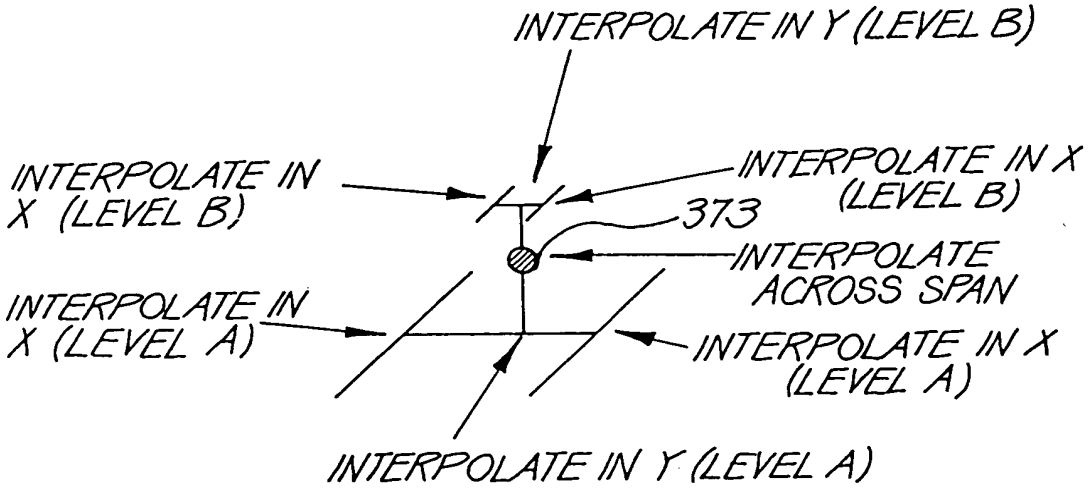

INTERPOLATE IN Y (LEVEL B)
INTERPOLATE IN X (LEVEL B)
INTERPOLATE IN X (LEVEL B)
373
INTERPOLATE ACROSS SPAN
INTERPOLATE IN X (LEVEL A)
INTERPOLATE IN X (LEVEL A)
INTERPOLATE IN Y (LEVEL A)

FIG. 91

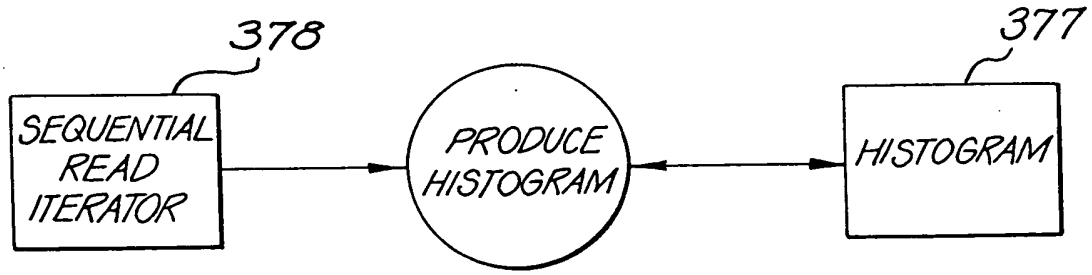

378
377
SEQUENTIAL READ ITERATOR
PRODUCE HISTOGRAM
HISTOGRAM

FIG. 92

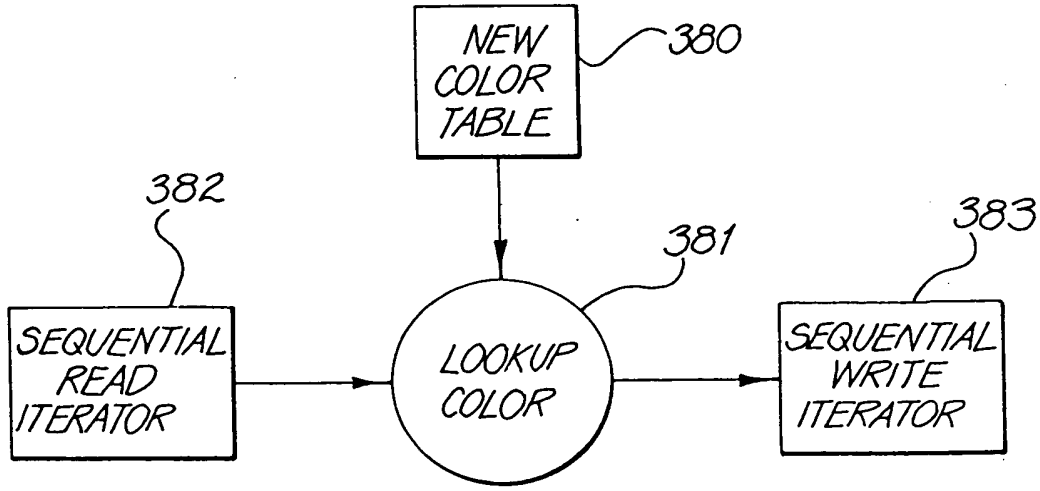

NEW COLOR TABLE
380
382
381
383
SEQUENTIAL READ ITERATOR
LOOKUP COLOR
SEQUENTIAL WRITE ITERATOR

FIG. 93

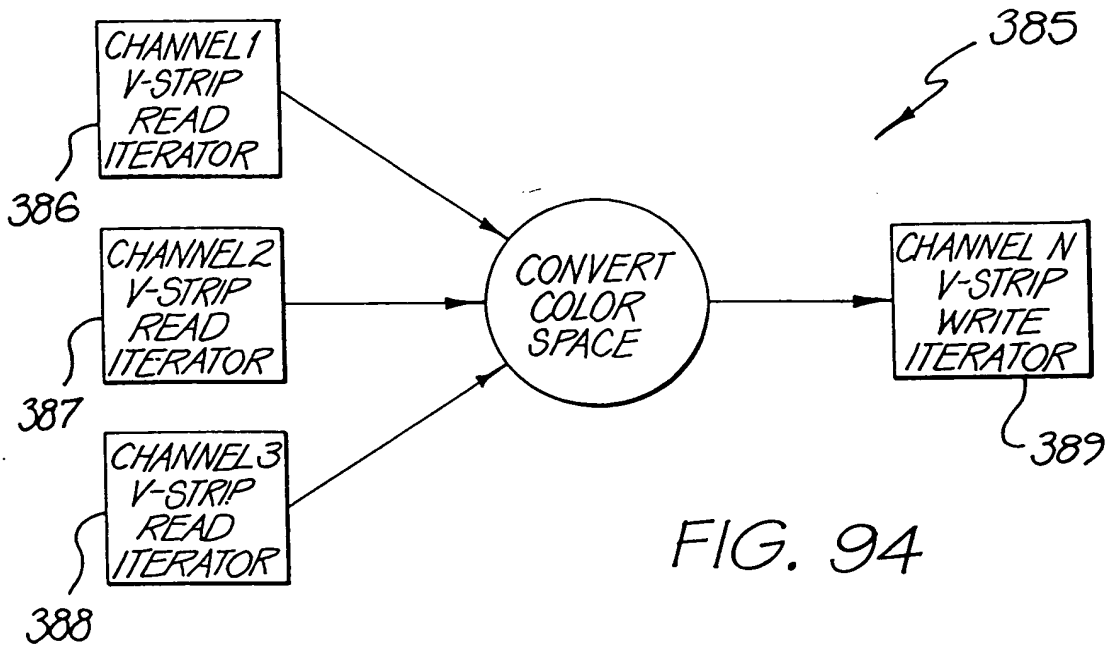


FIG. 94

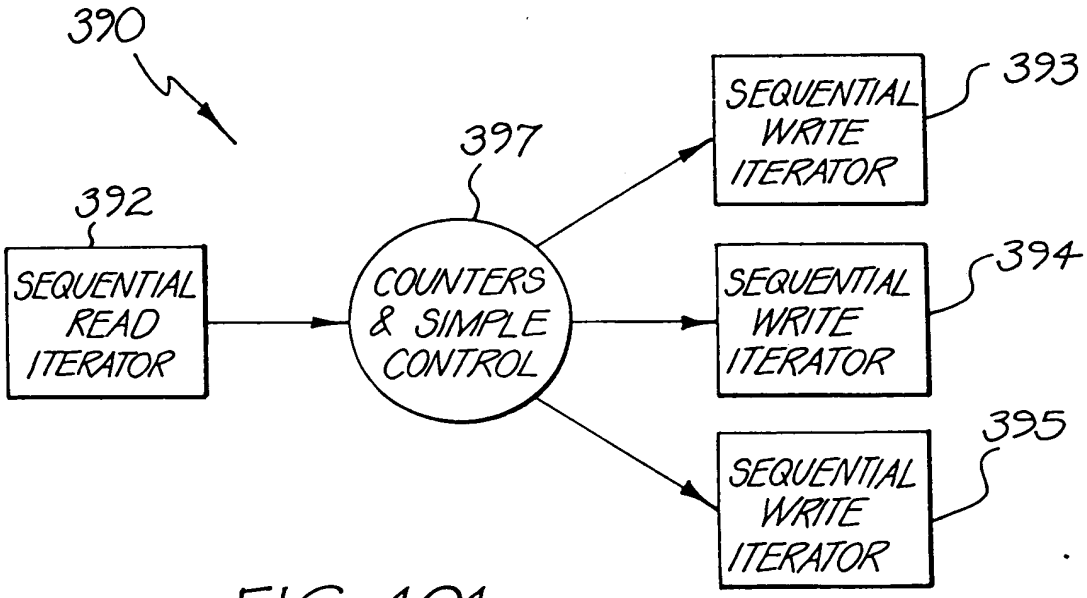


FIG. 101

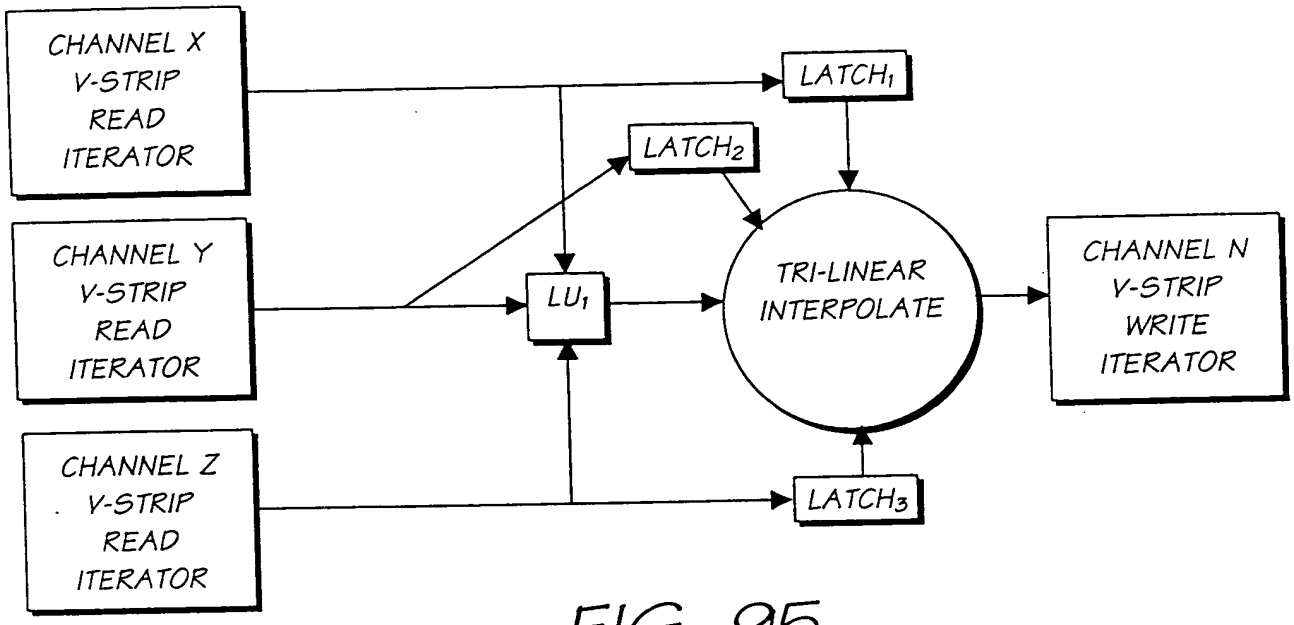

CHANNEL X V-STRIP READ ITERATOR
CHANNEL Y V-STRIP READ ITERATOR
CHANNEL Z V-STRIP READ ITERATOR
LATCH1
LATCH2
LATCH3
LU1
TRI-LINEAR INTERPOLATE
CHANNEL N V-STRIP WRITE ITERATOR

FIG. 95

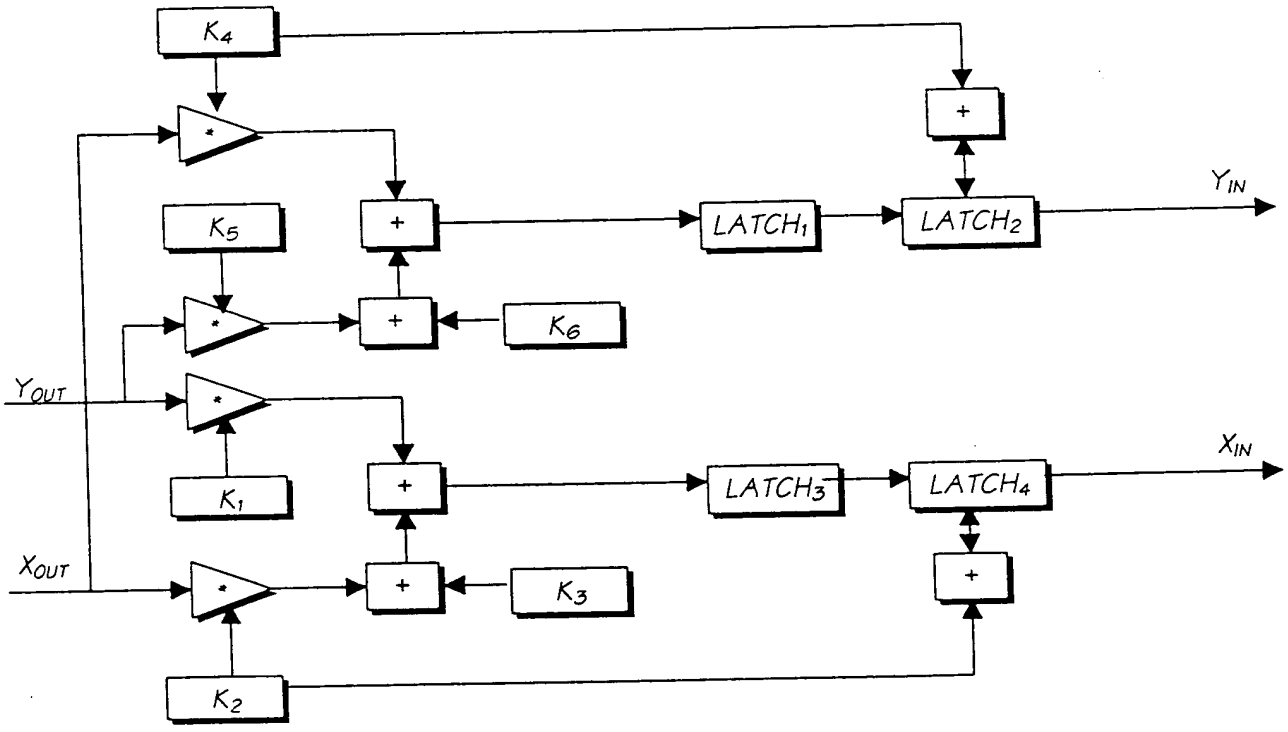

K4
K5
K6
K1
K2
K3
YOUT
XOUT
LATCH1
LATCH2
LATCH3
LATCH4
YIN
XIN

FIG. 96

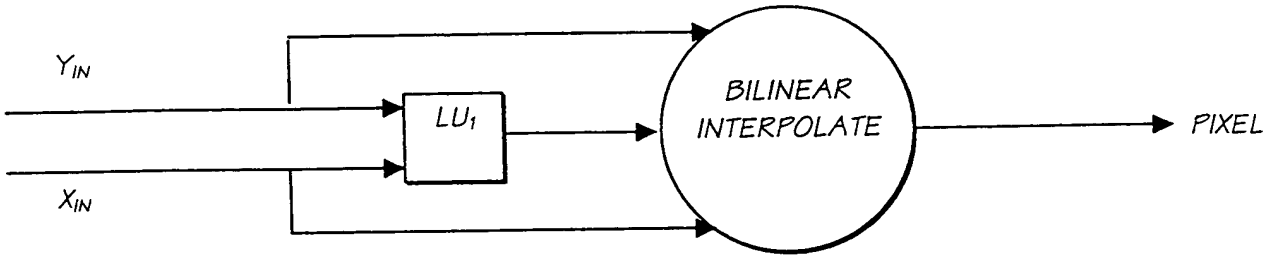

Y_{IN}
X_{IN}
LU_1
BILINEAR INTERPOLATE
PIXEL

FIG. 97

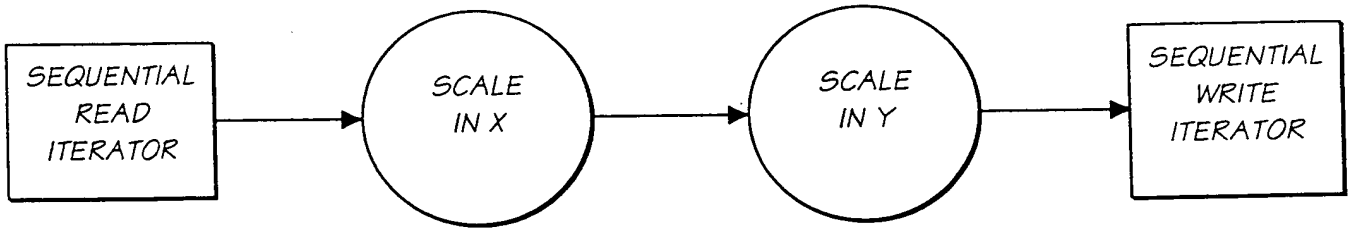

SEQUENTIAL READ ITERATOR
SCALE IN X
SCALE IN Y
SEQUENTIAL WRITE ITERATOR

FIG. 98

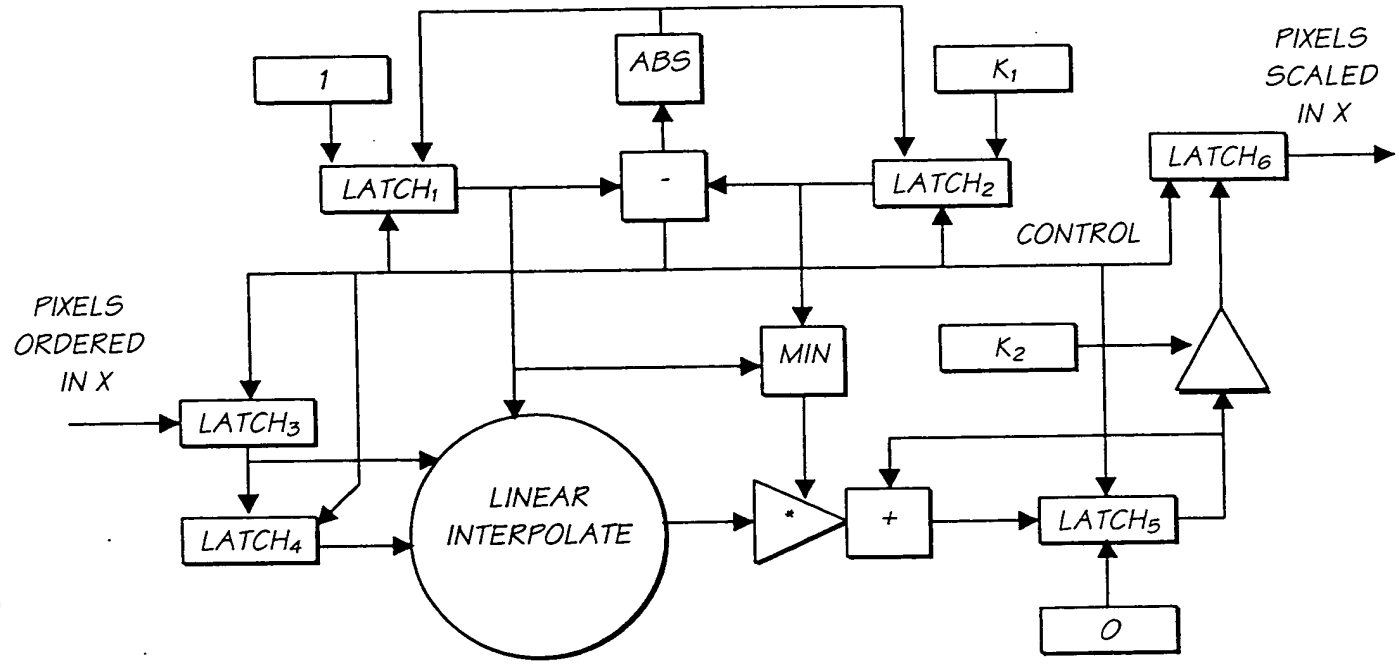


FIG. 99

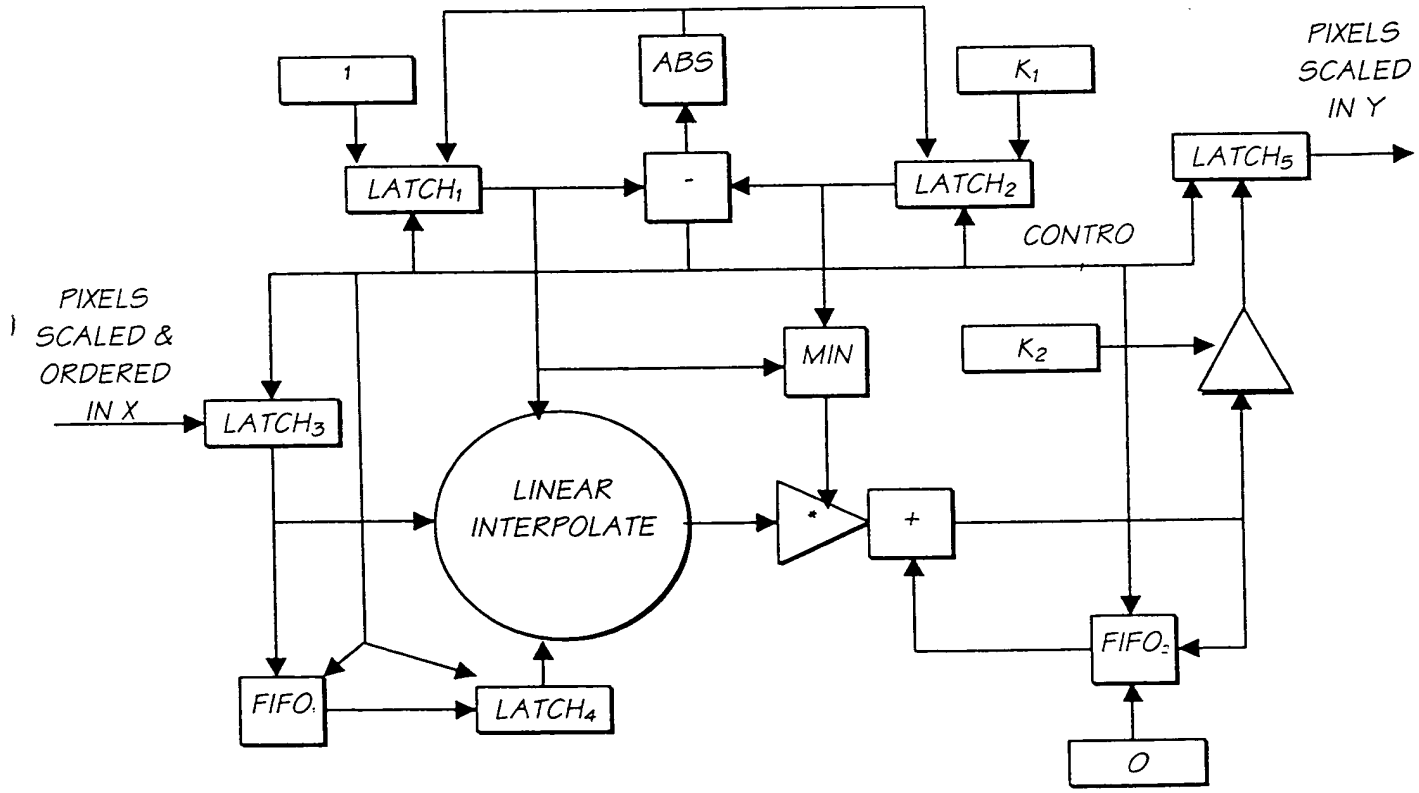


FIG. 100

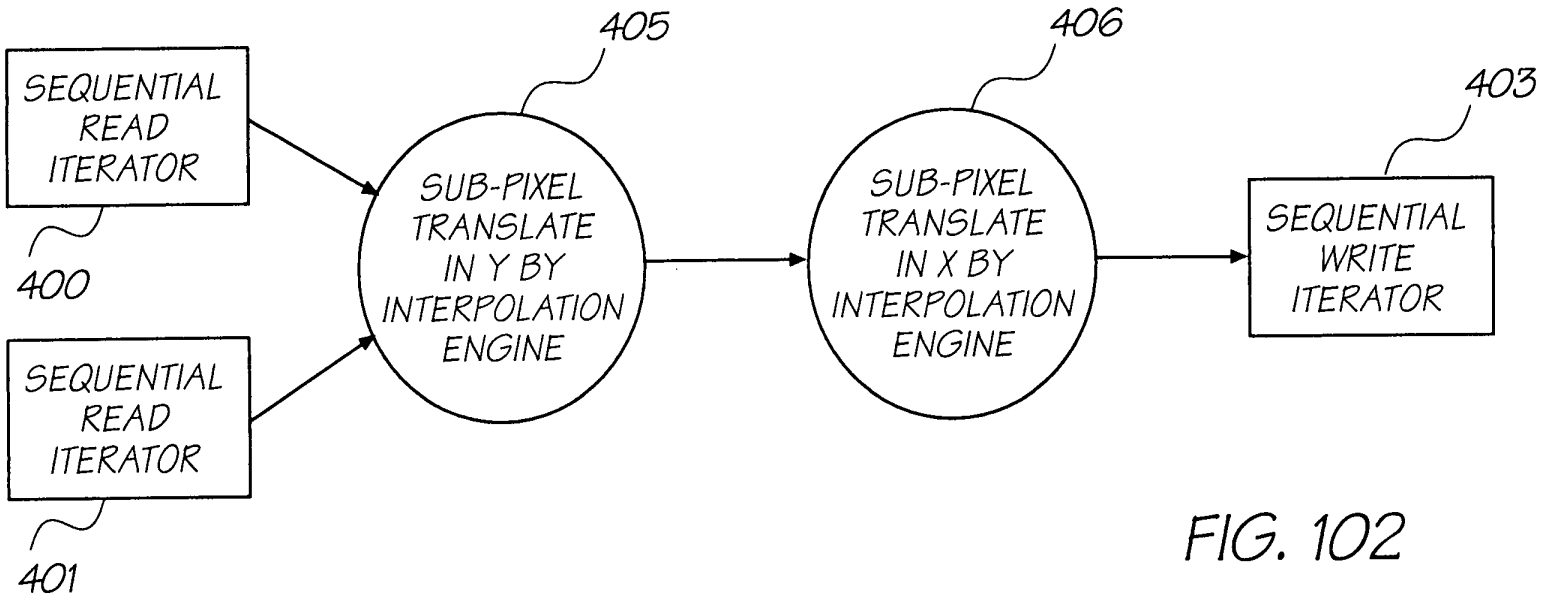


FIG. 102

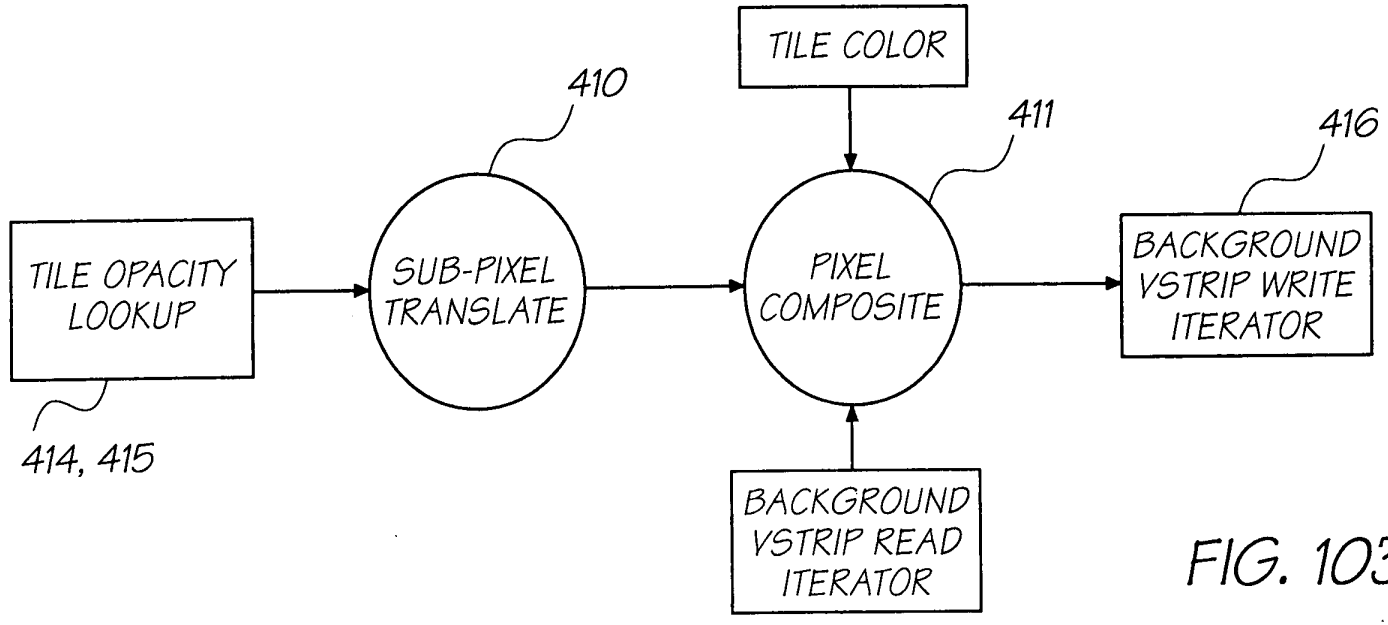


FIG. 103

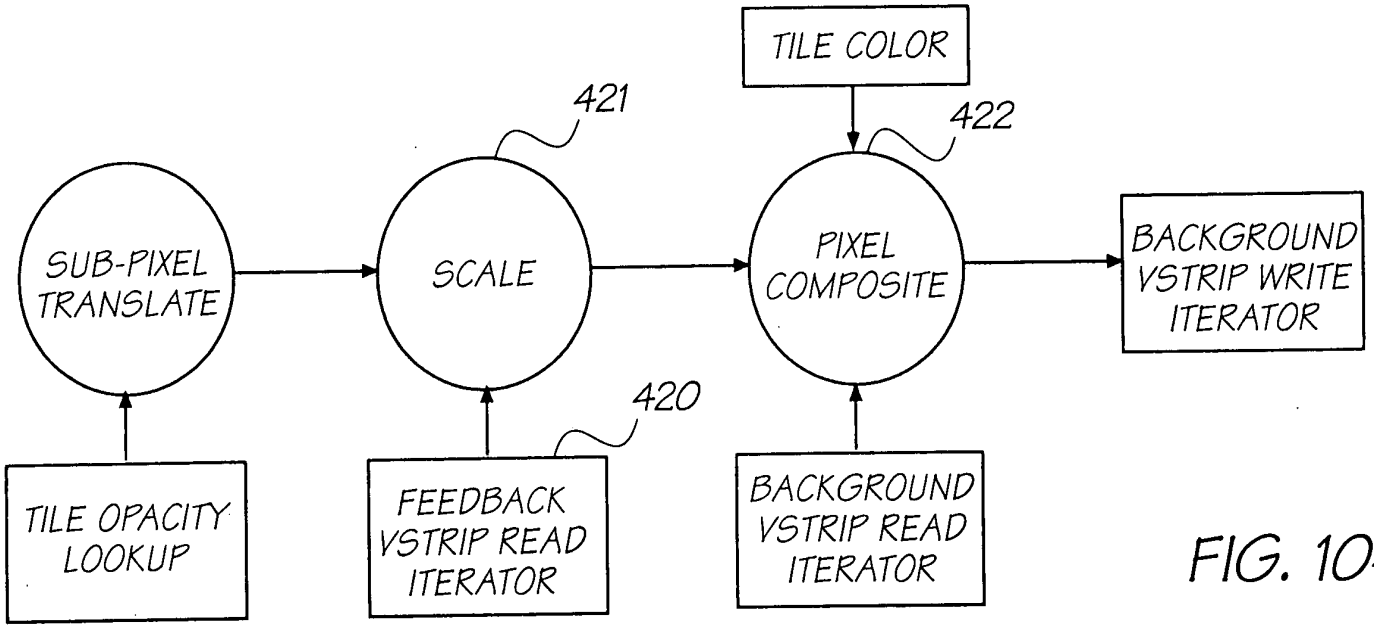


FIG. 104

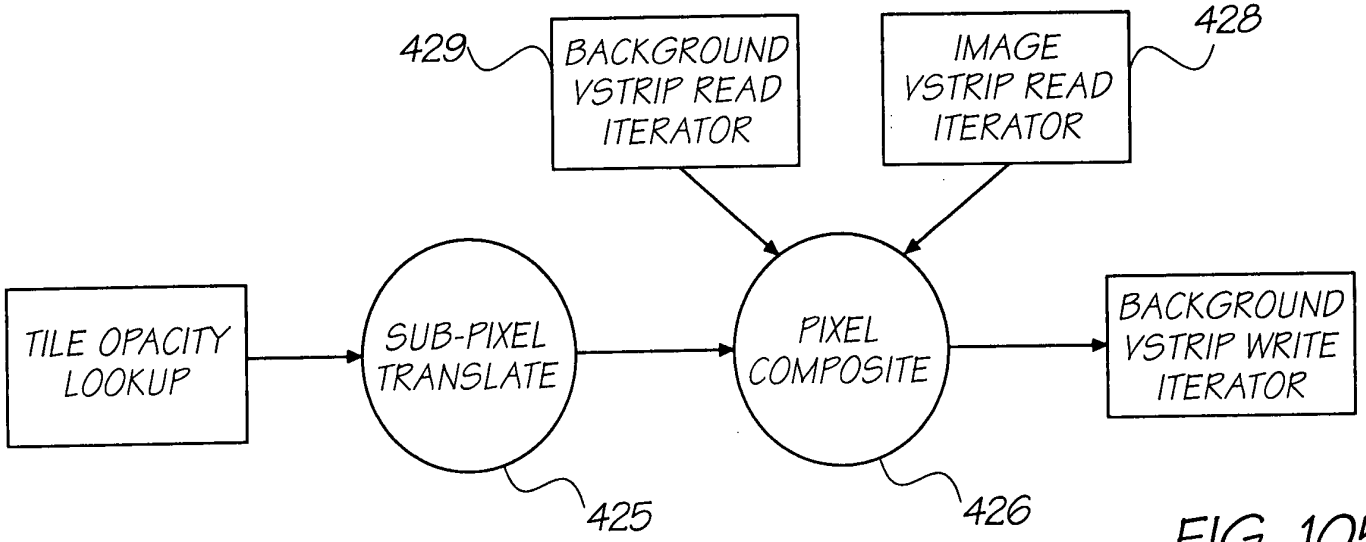


FIG. 105

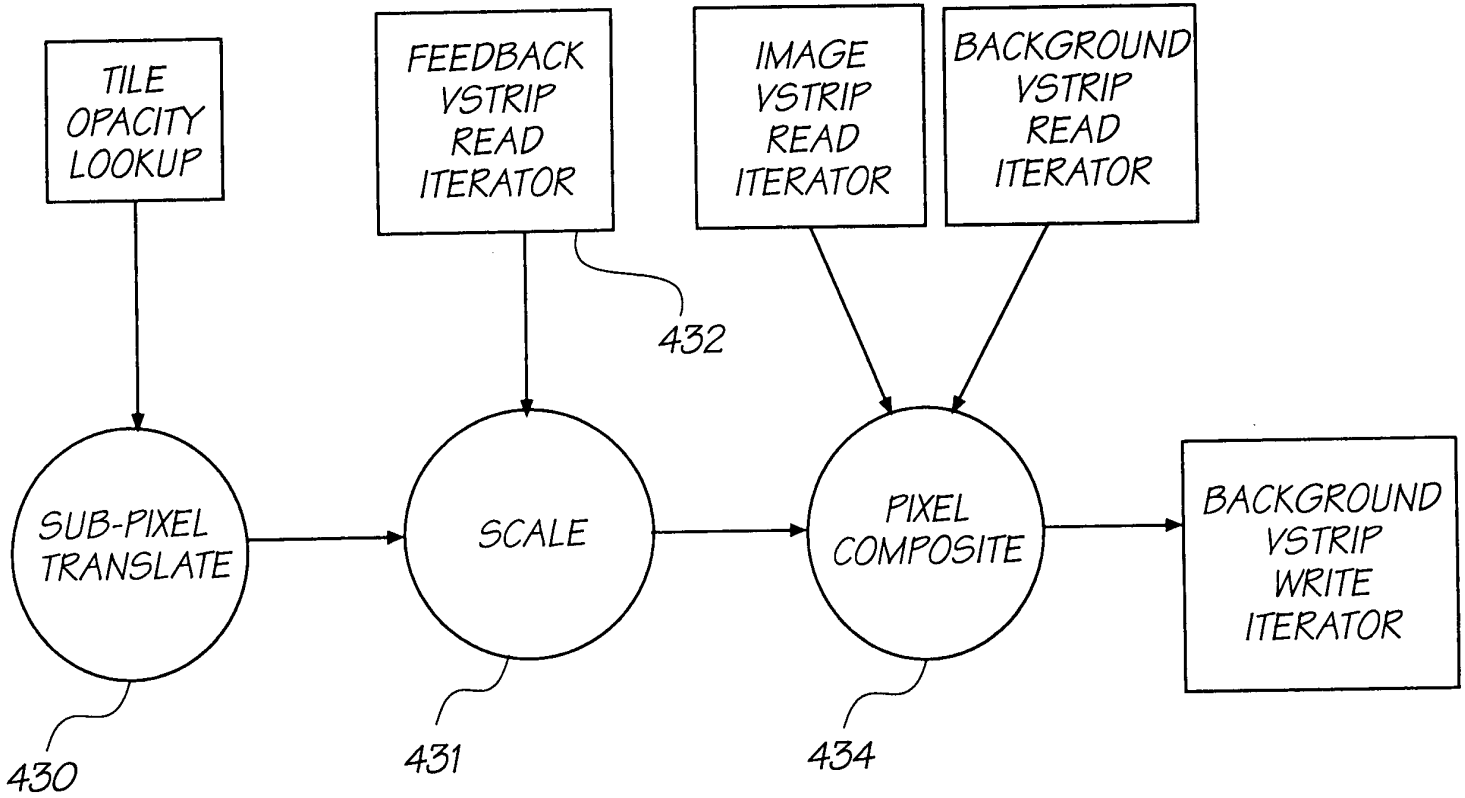

TILE OPACITY LOOKUP
FEEDBACK VSTRIP READ ITERATOR
IMAGE VSTRIP READ ITERATOR
BACKGROUND VSTRIP READ ITERATOR
432
SUB-PIXEL TRANSLATE
SCALE
PIXEL COMPOSITE
BACKGROUND VSTRIP WRITE ITERATOR
430
431
434

FIG. 106

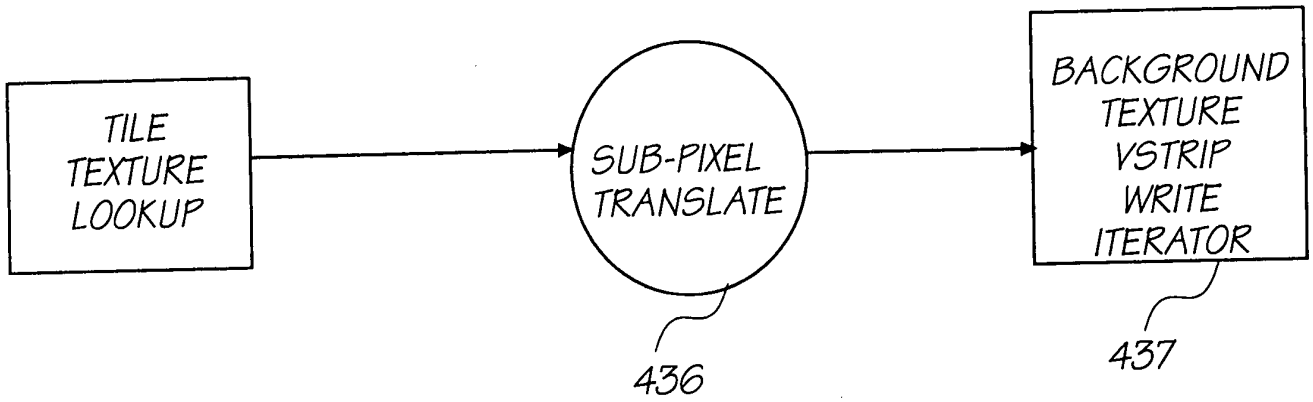

TILE TEXTURE LOOKUP
SUB-PIXEL TRANSLATE
BACKGROUND TEXTURE VSTRIP WRITE ITERATOR
436
437

FIG. 107

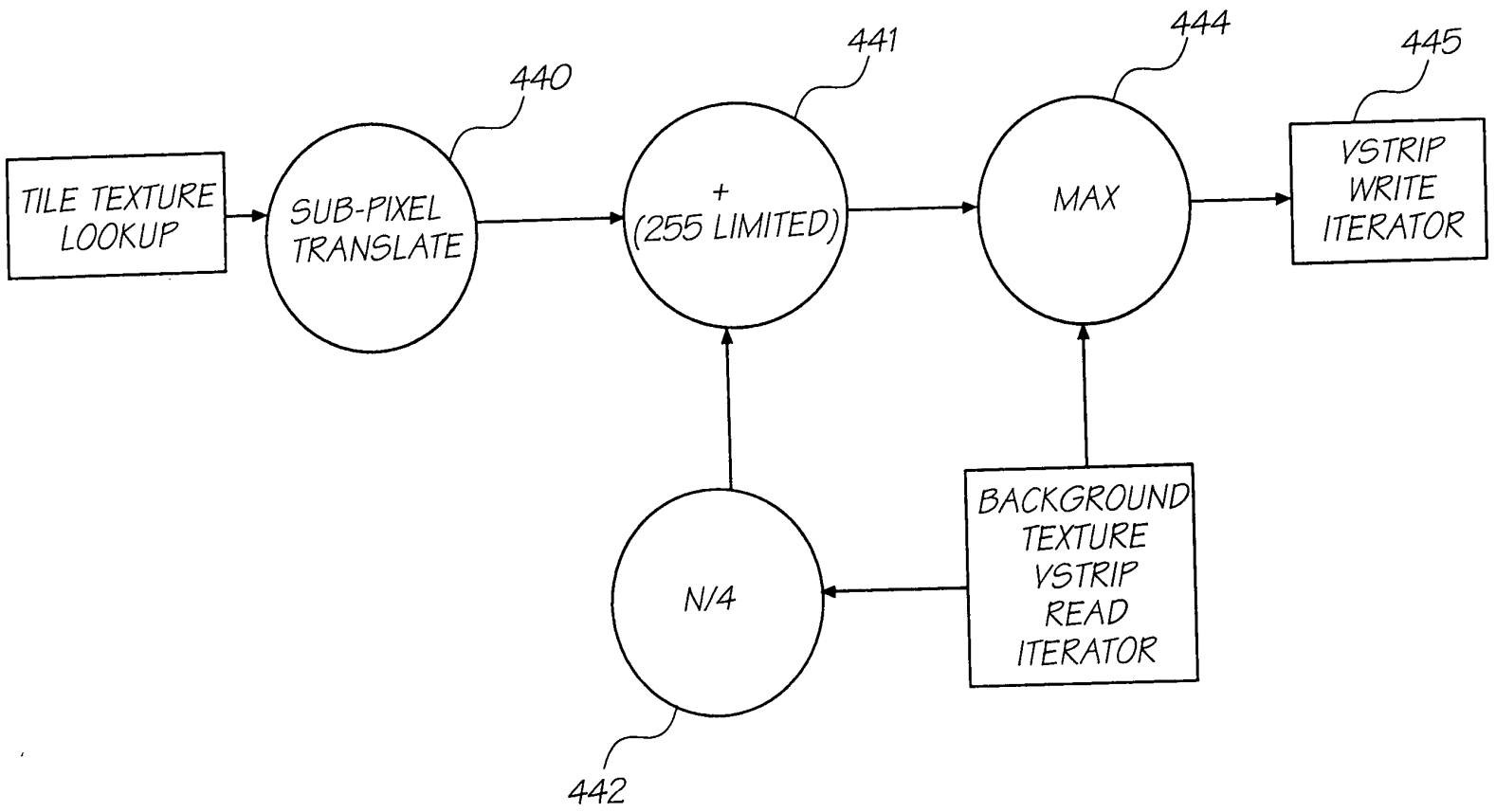

TILE TEXTURE LOOKUP
SUB-PIXEL TRANSLATE
440
+ (255 LIMITED)
441
MAX
444
VSTRIP WRITE ITERATOR
445
N/4
442
BACKGROUND TEXTURE VSTRIP READ ITERATOR

FIG. 108

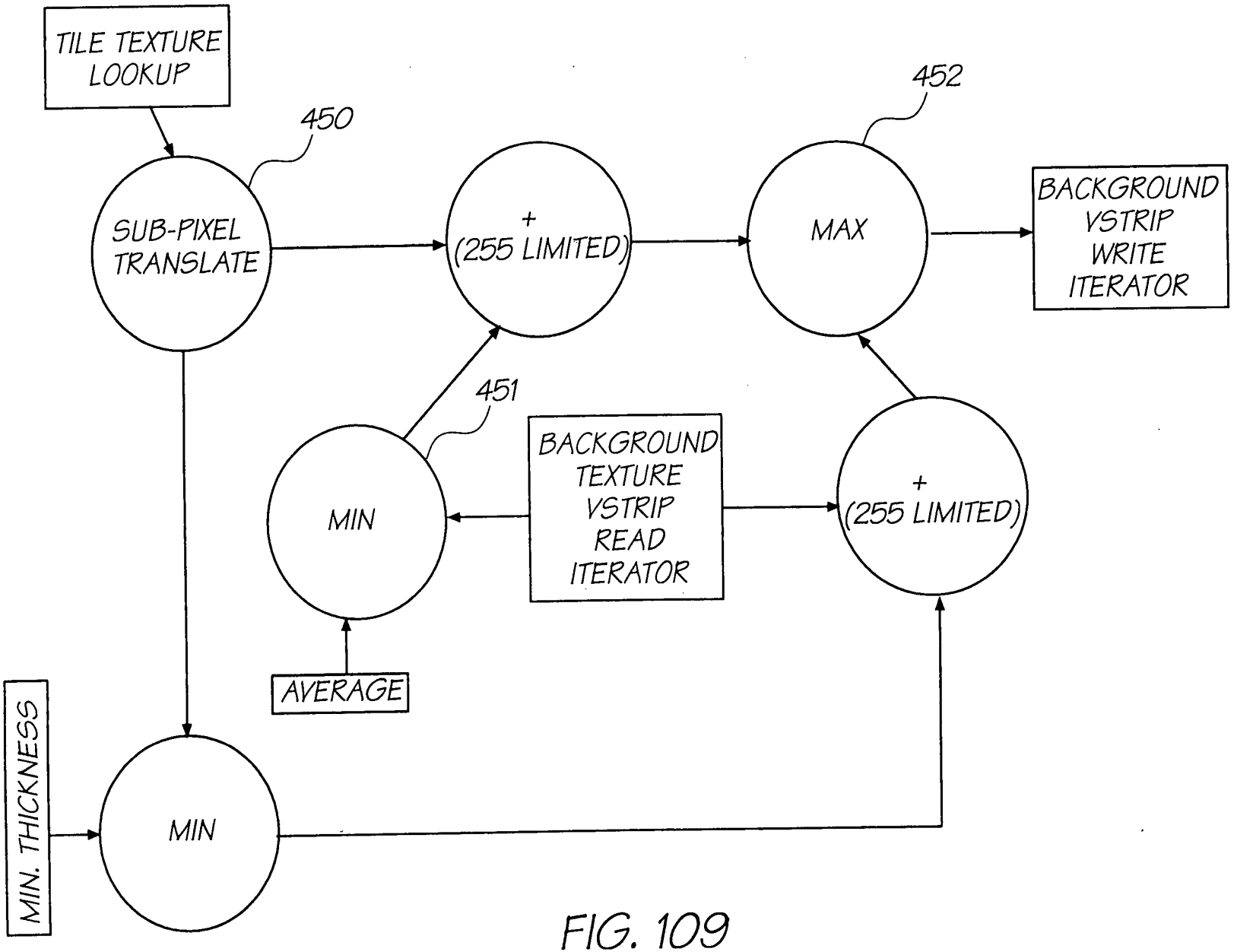

TILE TEXTURE LOOKUP
450
SUB-PIXEL TRANSLATE
+ (255 LIMITED)
452
MAX
BACKGROUND VSTRIP WRITE ITERATOR
451
MIN
BACKGROUND TEXTURE VSTRIP READ ITERATOR
+ (255 LIMITED)
AVERAGE
MIN. THICKNESS
MIN

FIG. 109

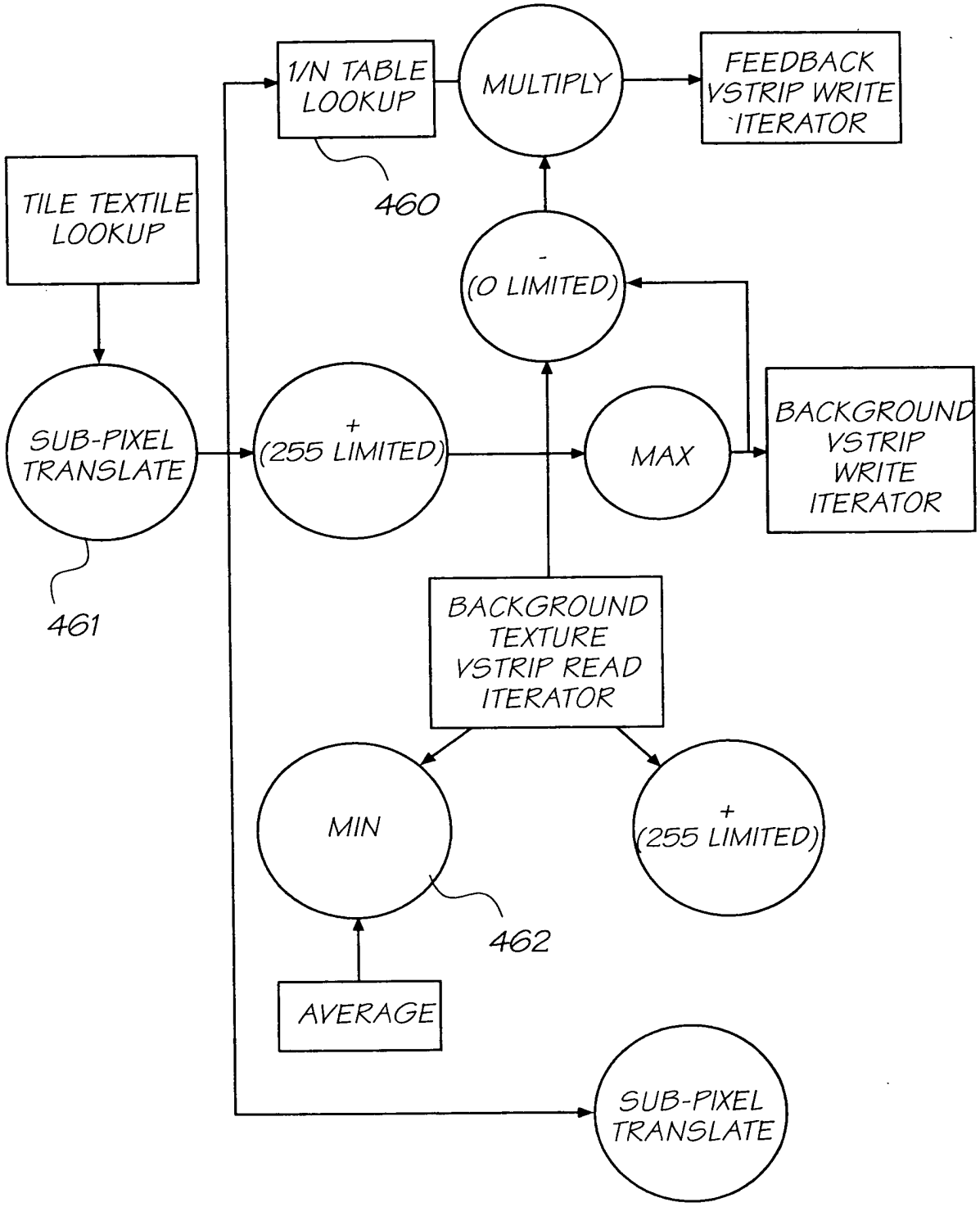

1/N TABLE LOOKUP
MULTIPLY
FEEDBACK VSTRIP WRITE ITERATOR
460
TILE TEXTILE LOOKUP
- (0 LIMITED)
SUB-PIXEL TRANSLATE
+ (255 LIMITED)
MAX
BACKGROUND VSTRIP WRITE ITERATOR
461
BACKGROUND TEXTURE VSTRIP READ ITERATOR
MIN
+ (255 LIMITED)
462
AVERAGE
SUB-PIXEL TRANSLATE

FIG. 110

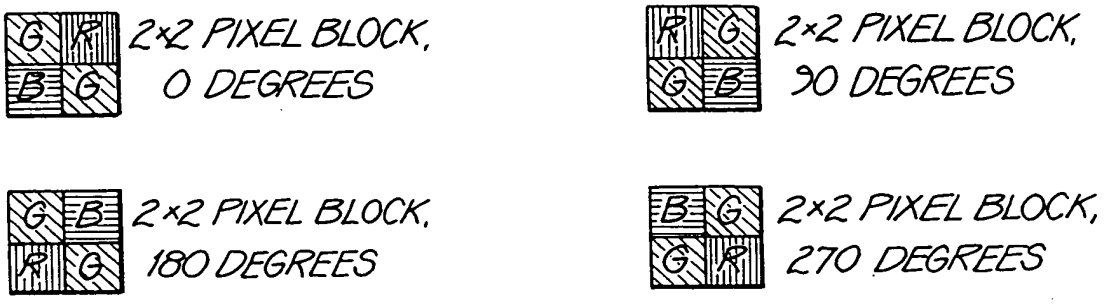


FIG. 111

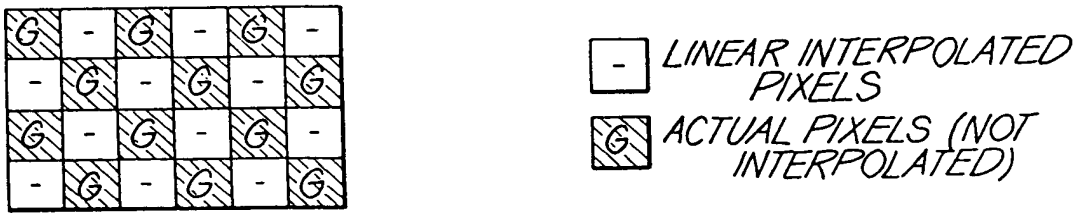


FIG. 112

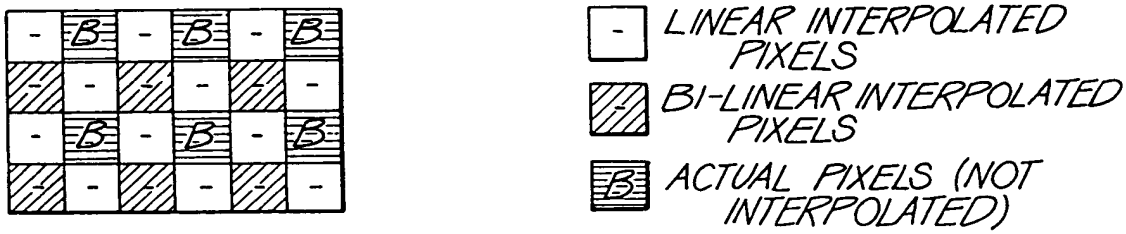


FIG. 113

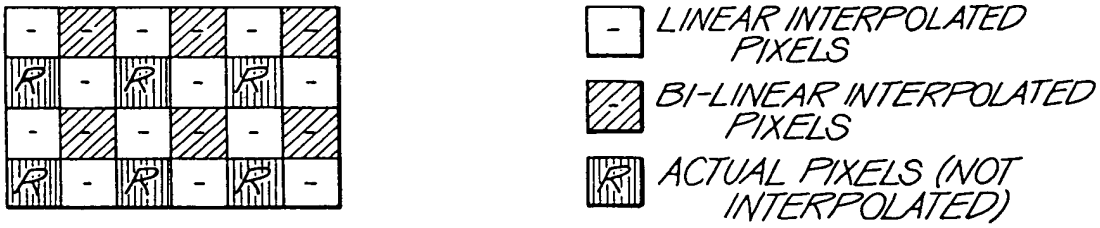


FIG. 114

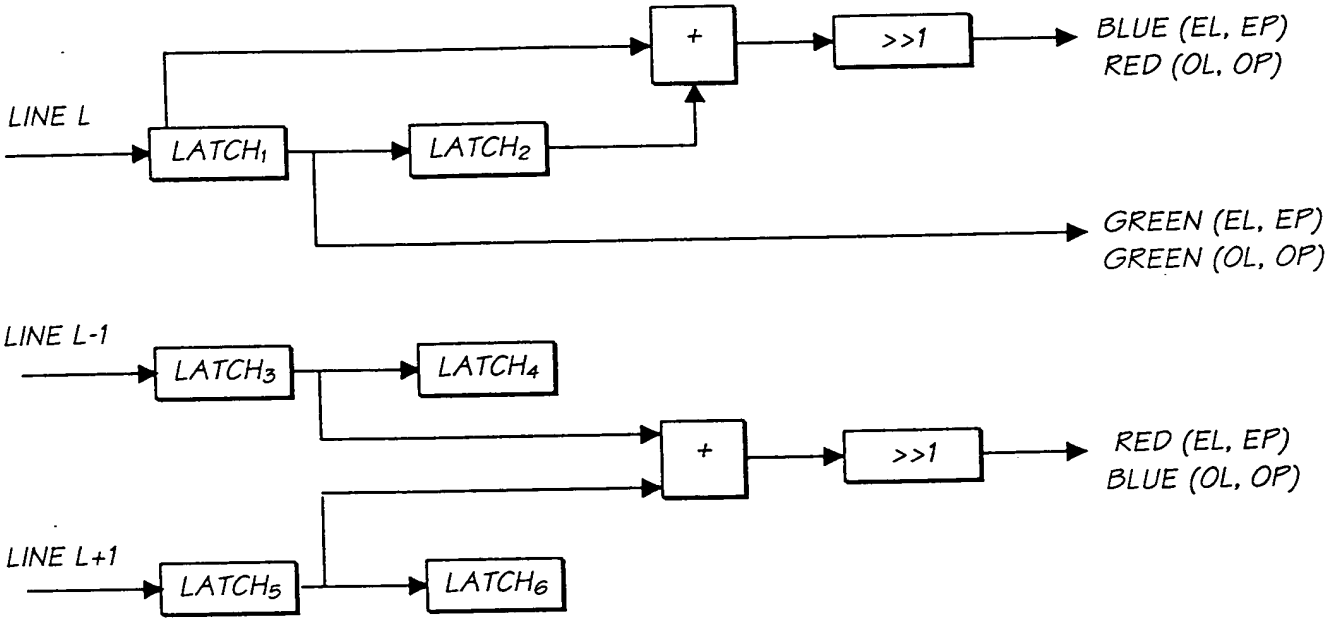


FIG. 115

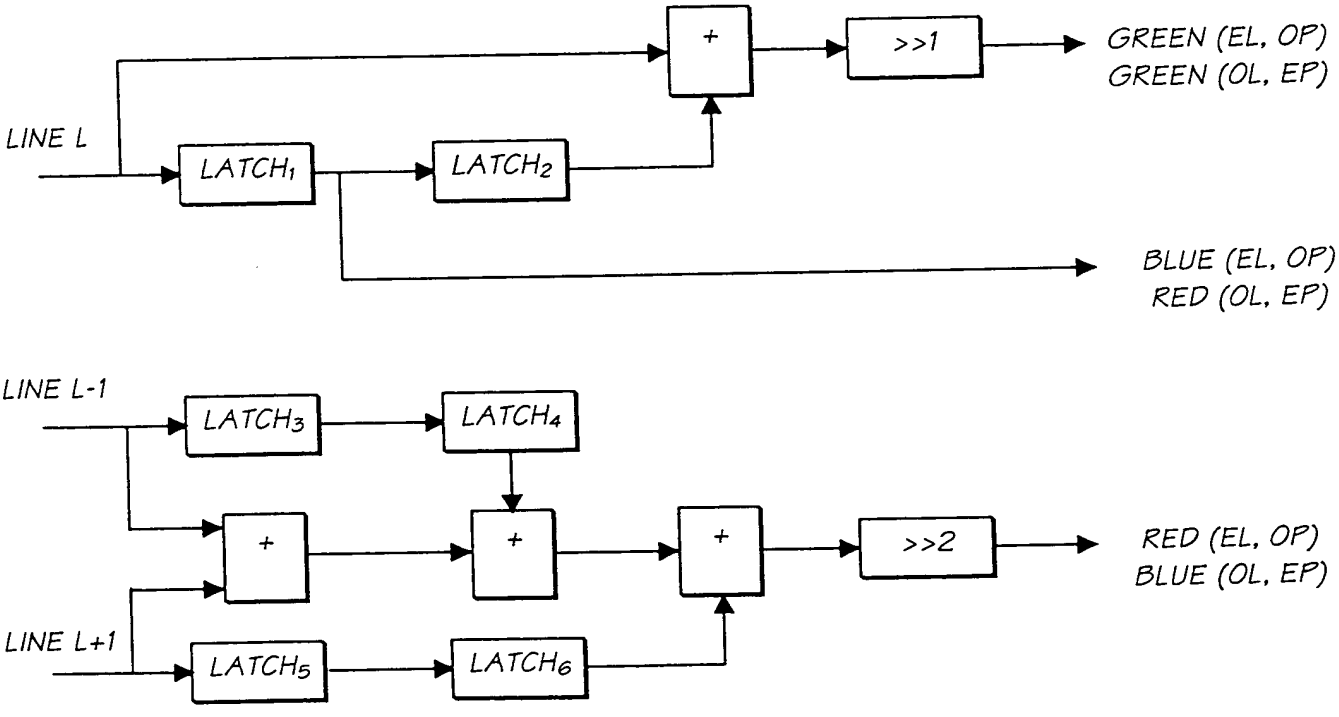


FIG. 116

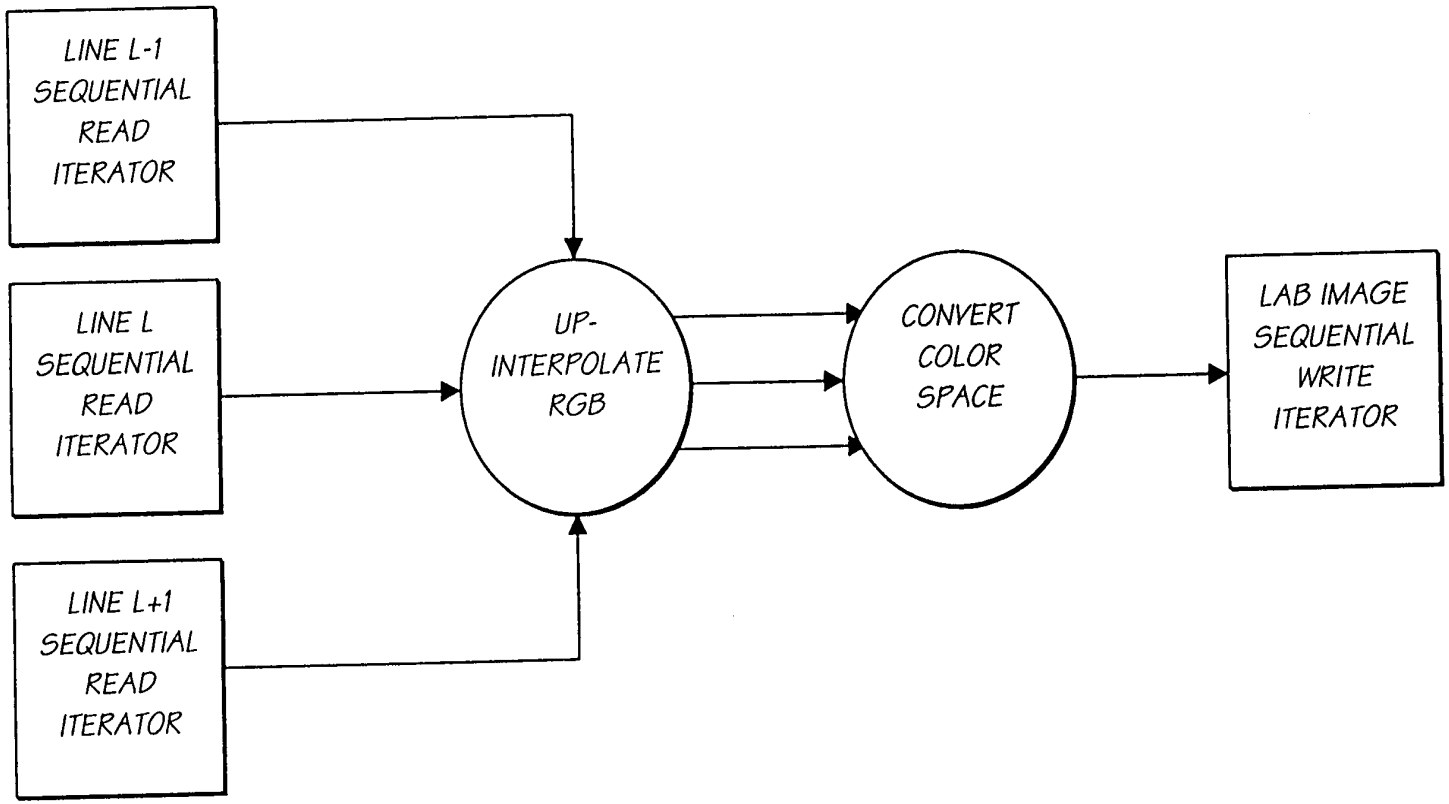

LINE L-1 SEQUENTIAL READ ITERATOR
LINE L SEQUENTIAL READ ITERATOR
LINE L+1 SEQUENTIAL READ ITERATOR
UP-INTERPOLATE RGB
CONVERT COLOR SPACE
LAB IMAGE SEQUENTIAL WRITE ITERATOR

FIG. 117

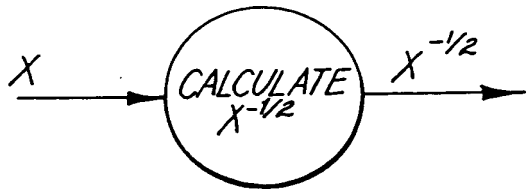


FIG. 118

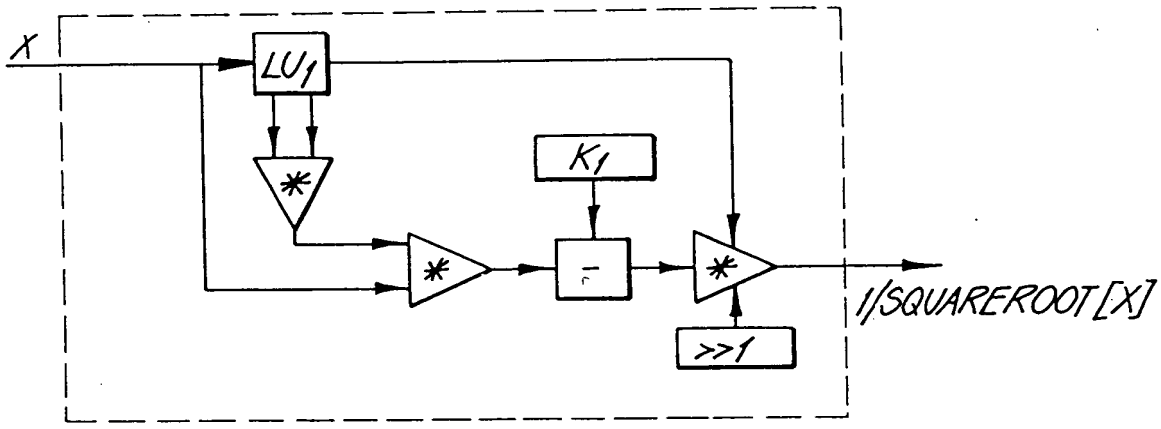


FIG. 119

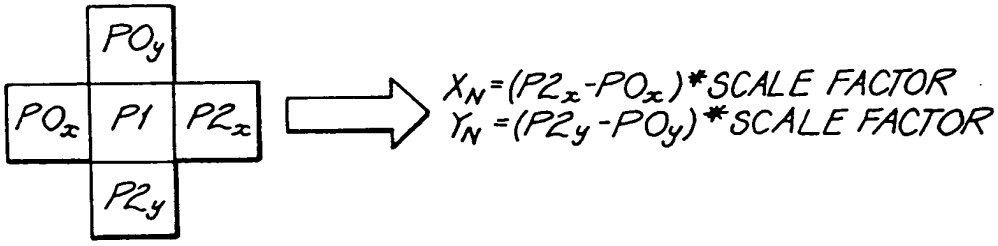


FIG. 120

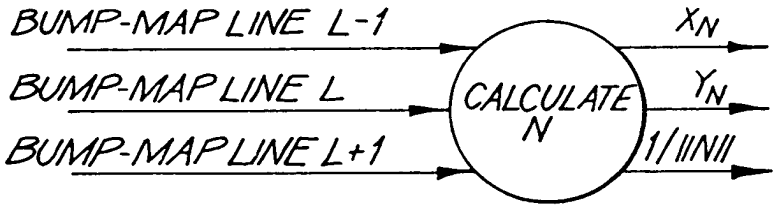


FIG. 121

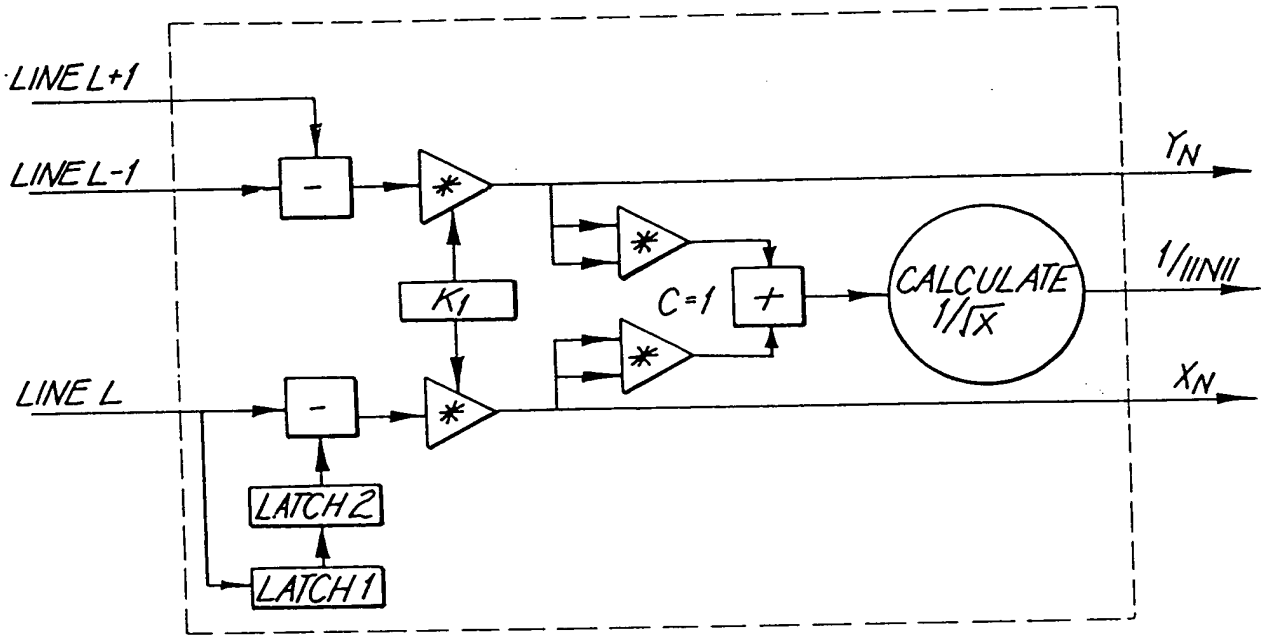

LINE L+1
LINE L-1
LINE L
K1
C=1
CALCULATE
1/√X
YN
1/IINII
XN
LATCH 2
LATCH 1

FIG. 122

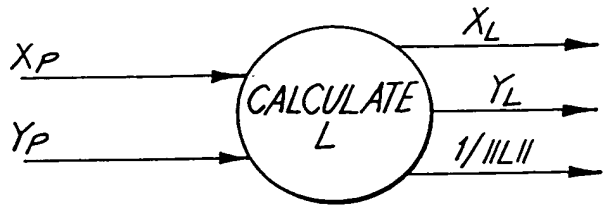


FIG. 123

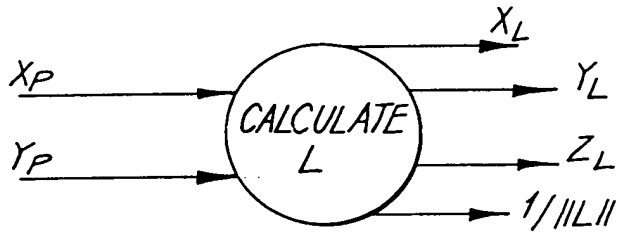


FIG. 124

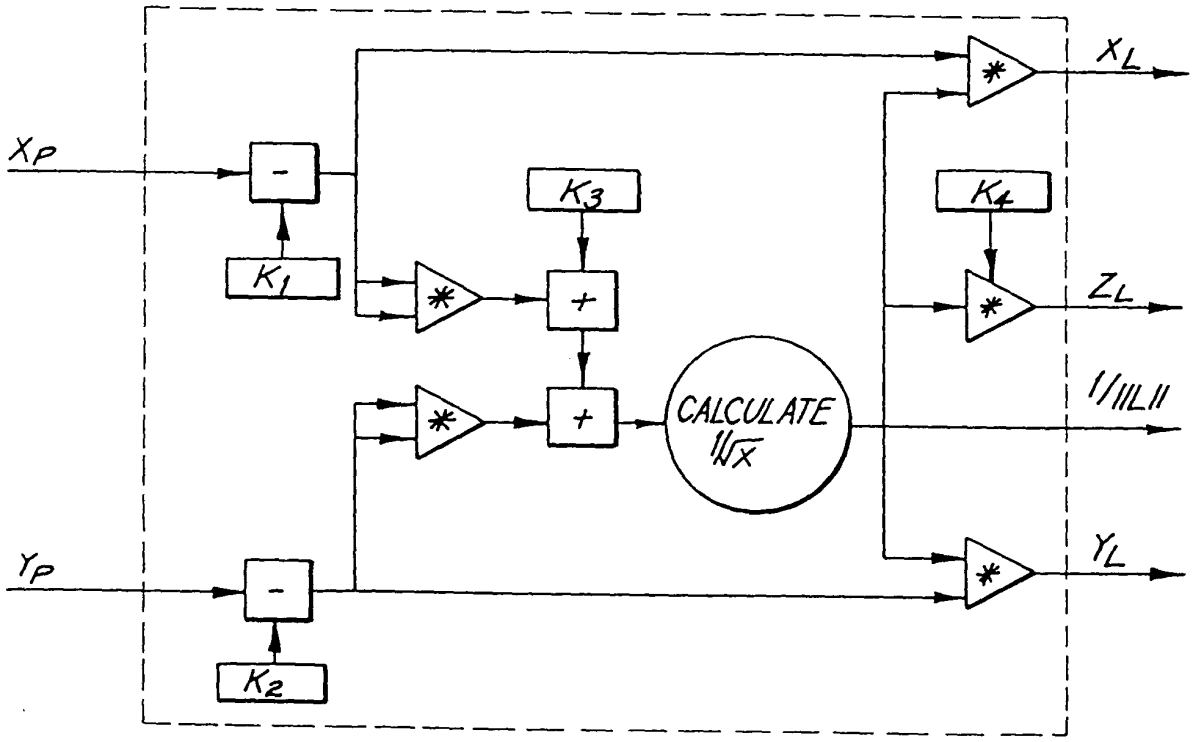


FIG. 125

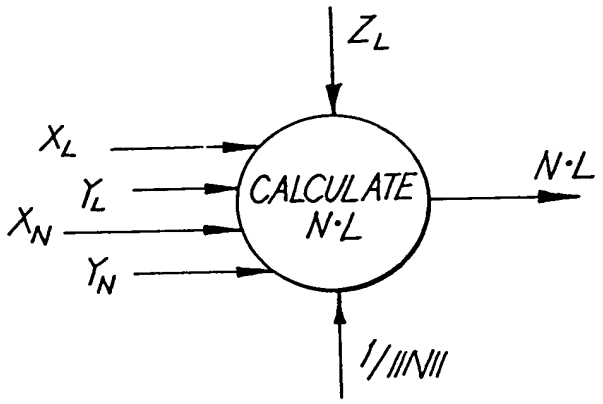


FIG. 126

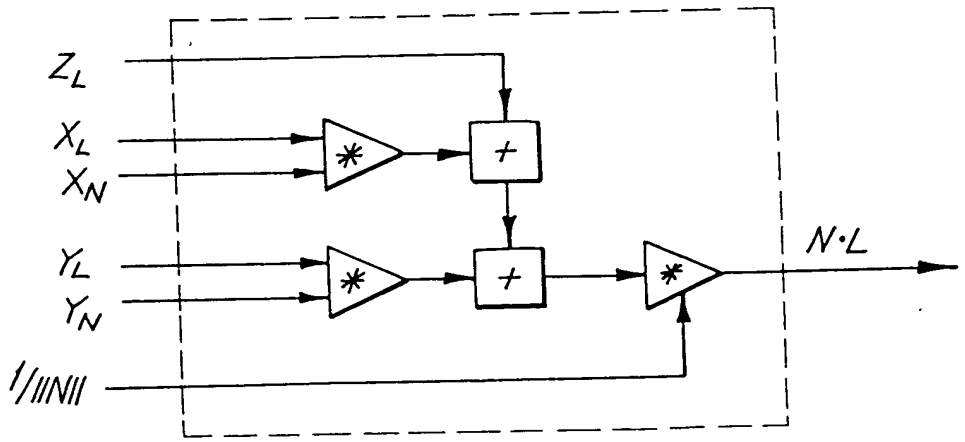


FIG. 127

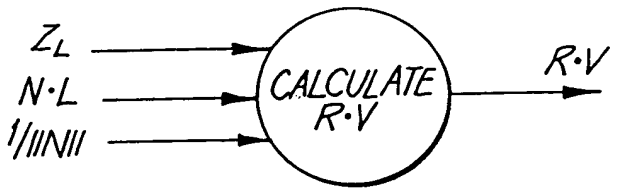


FIG. 128

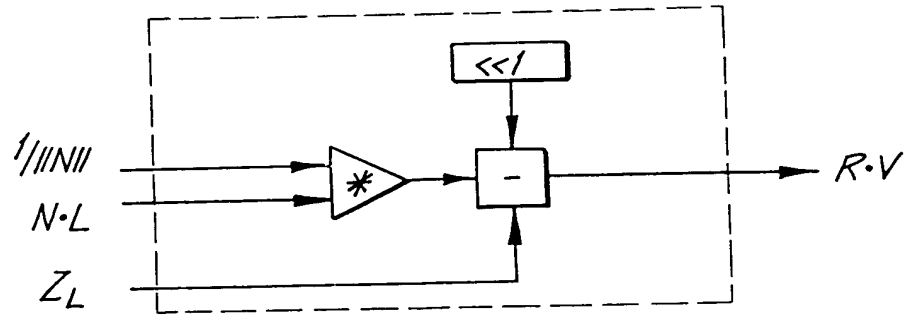


FIG. 129

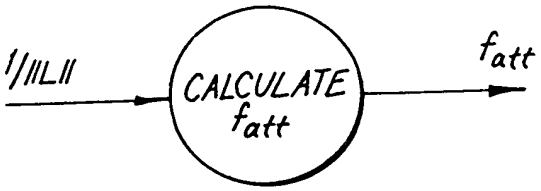


FIG. 130

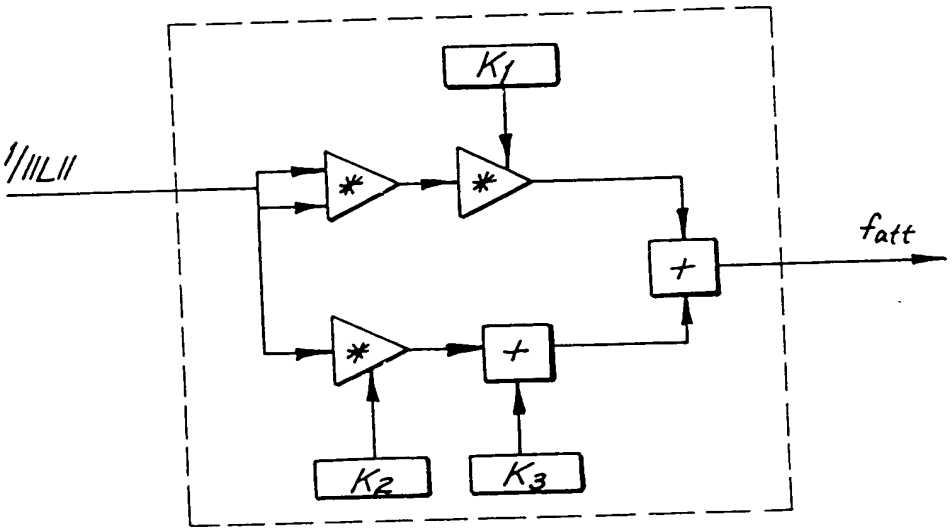


FIG. 131

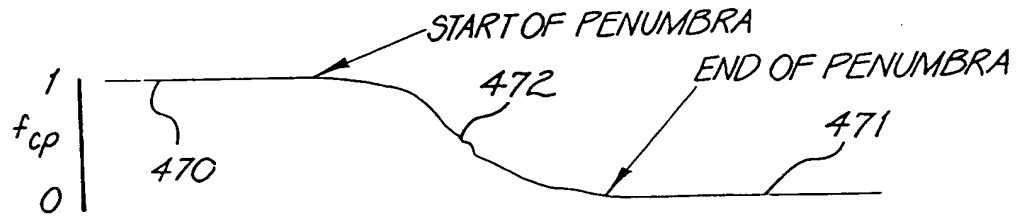


FIG. 132

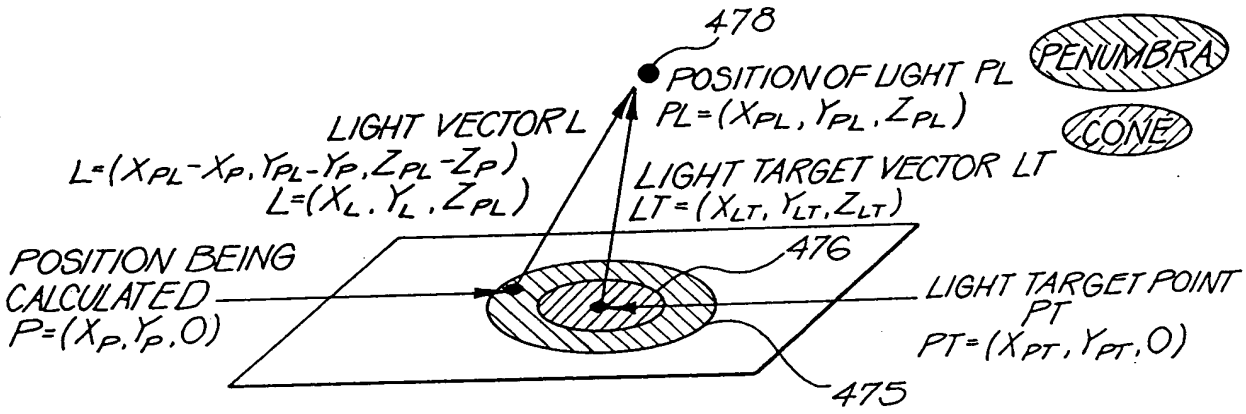


FIG. 133

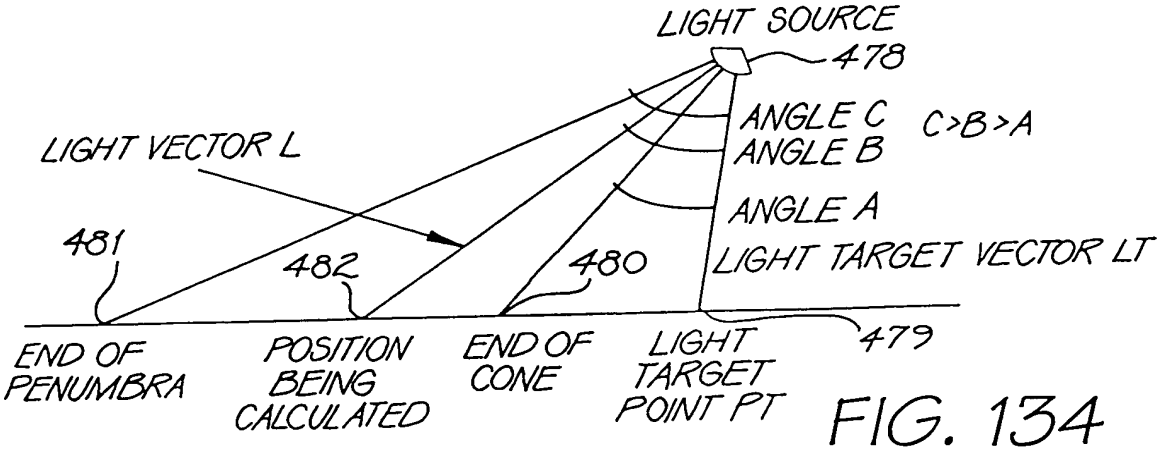


FIG. 134

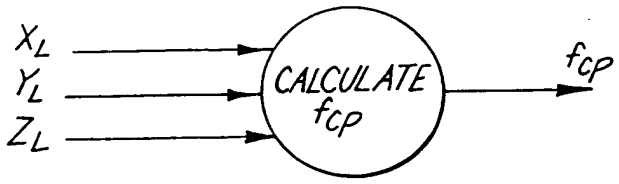


FIG. 135

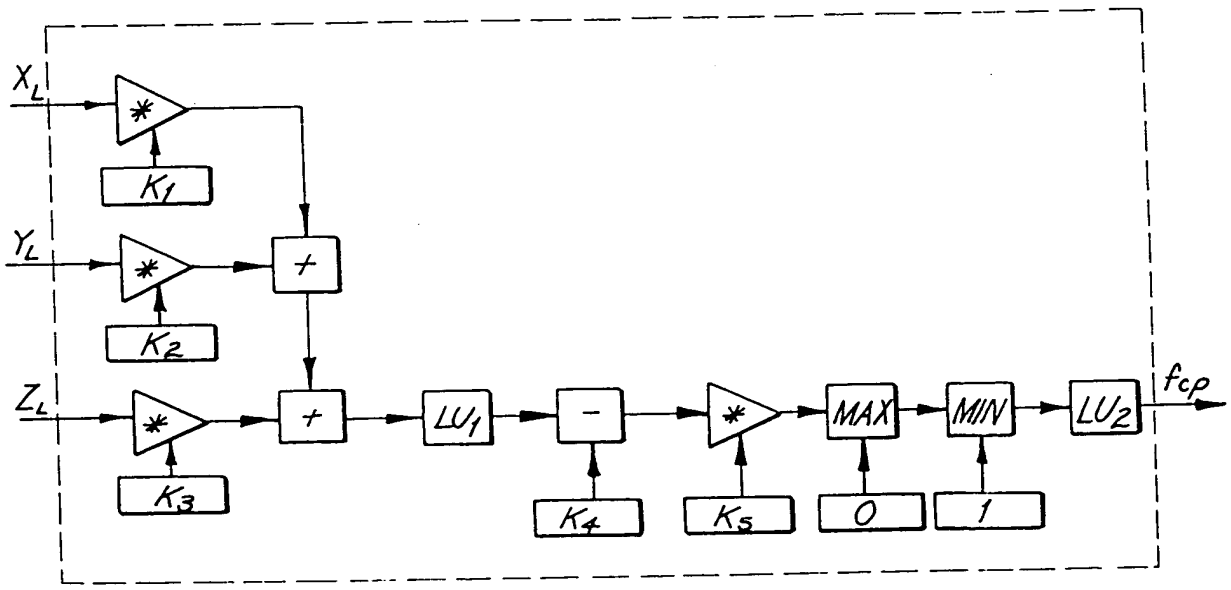


FIG. 136

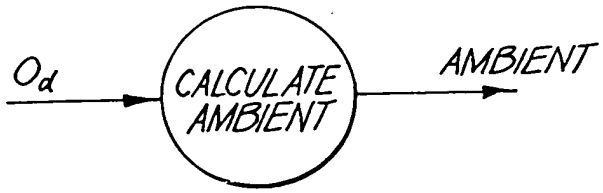


FIG. 137

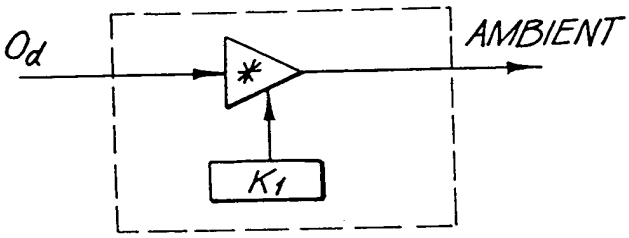


FIG. 138

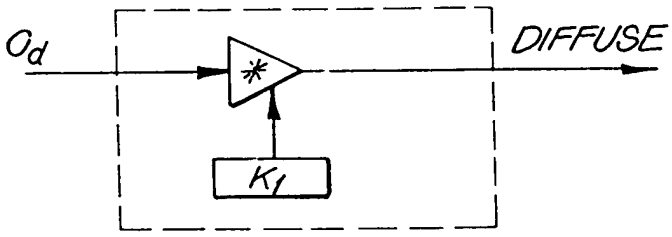


FIG. 139

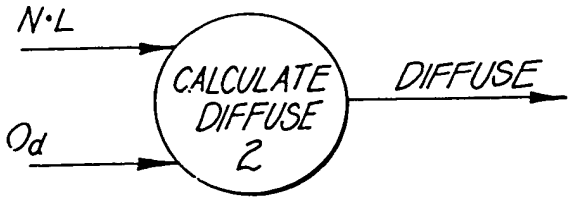


FIG. 140

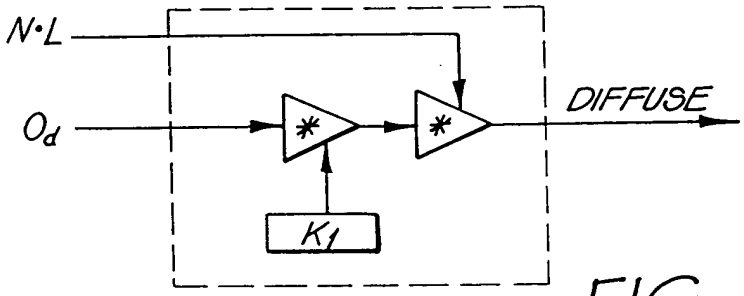


FIG. 141

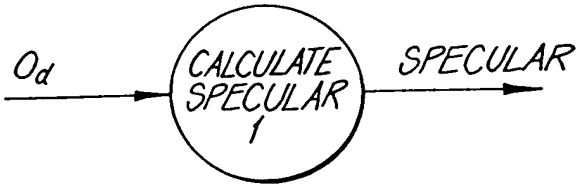


FIG. 142

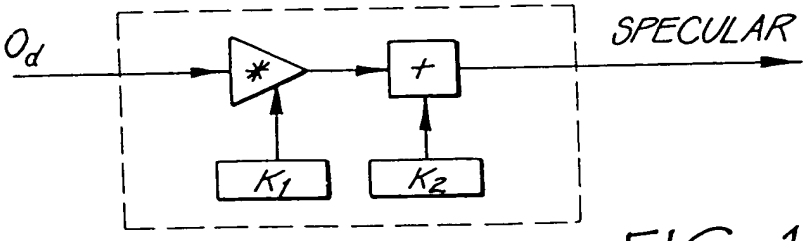


FIG. 143

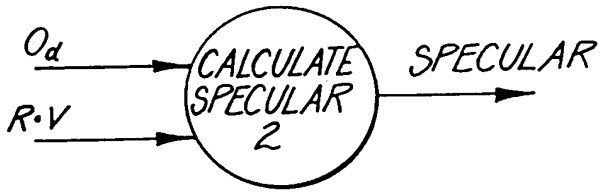


FIG. 144

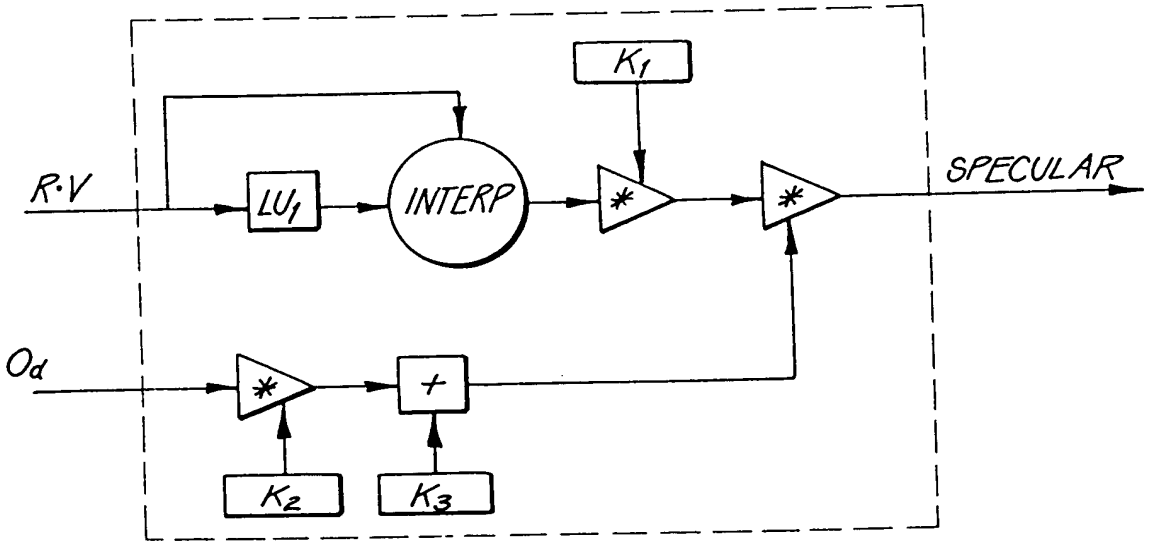


FIG. 145

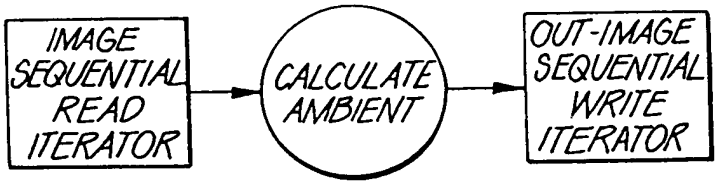


FIG. 146

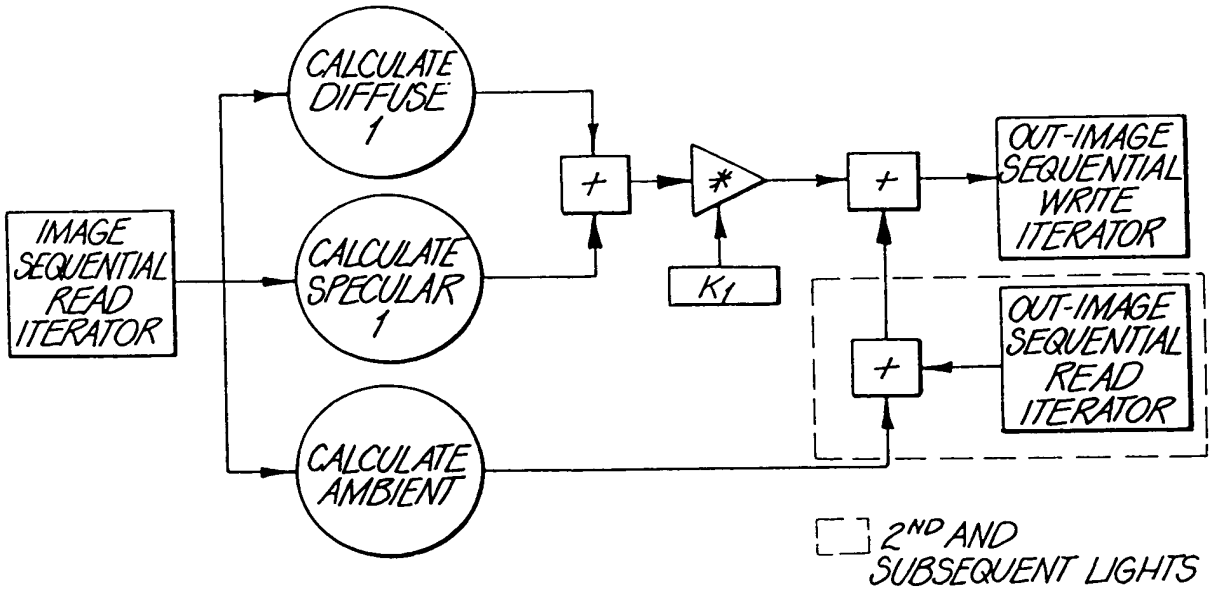


FIG. 147

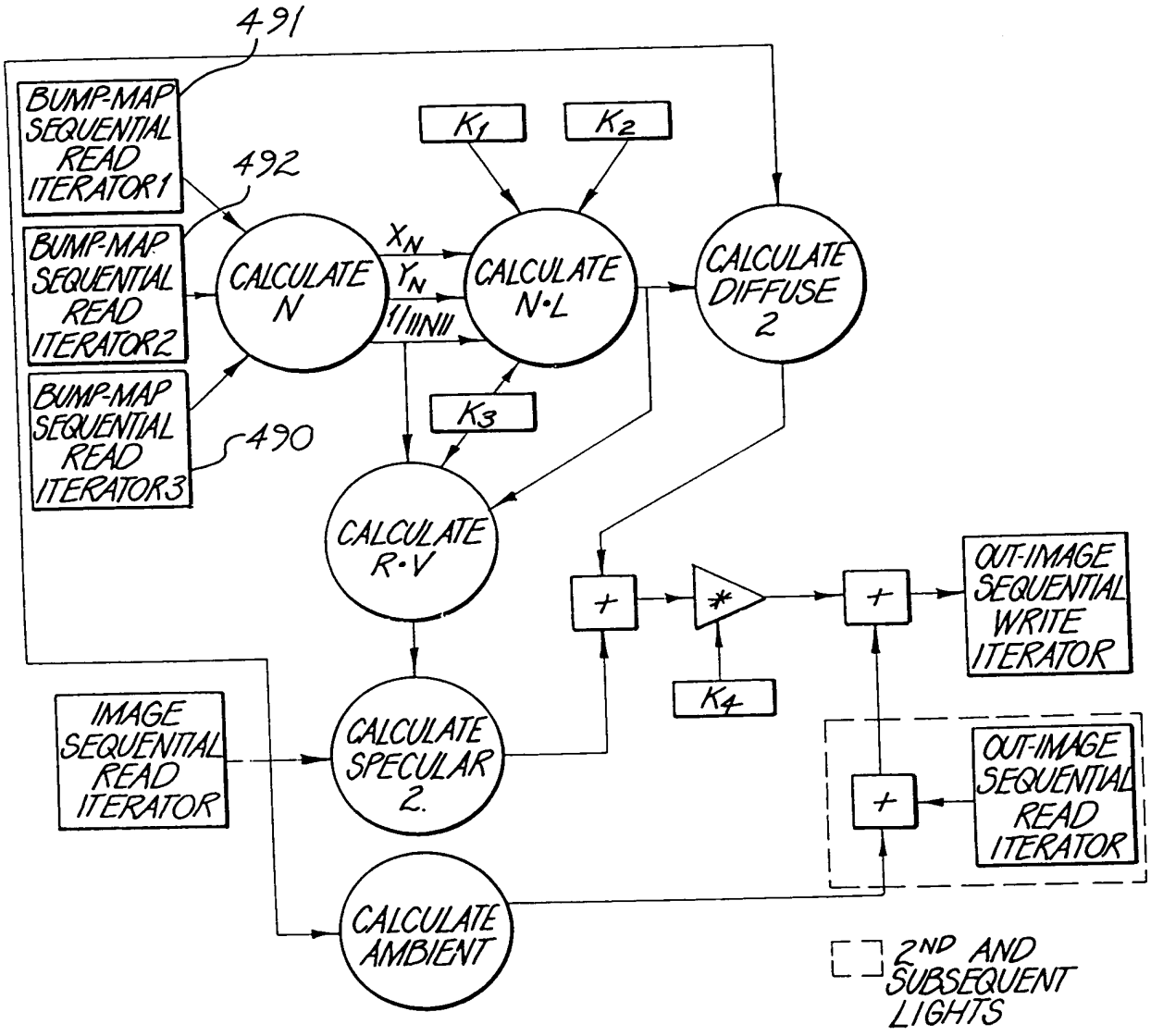

491
BUMP-MAP SEQUENTIAL READ ITERATOR1
492
BUMP-MAP SEQUENTIAL READ ITERATOR2
BUMP-MAP SEQUENTIAL READ ITERATOR3
490
K1
K2
CALCULATE N
XN
YN
1/||N||
CALCULATE N·L
CALCULATE DIFFUSE 2
K3
CALCULATE R·V
+
*
+
OUT-IMAGE SEQUENTIAL WRITE ITERATOR
K4
IMAGE SEQUENTIAL READ ITERATOR
CALCULATE SPECULAR 2.
+
OUT-IMAGE SEQUENTIAL READ ITERATOR
CALCULATE AMBIENT
2ND AND SUBSEQUENT LIGHTS

FIG. 148

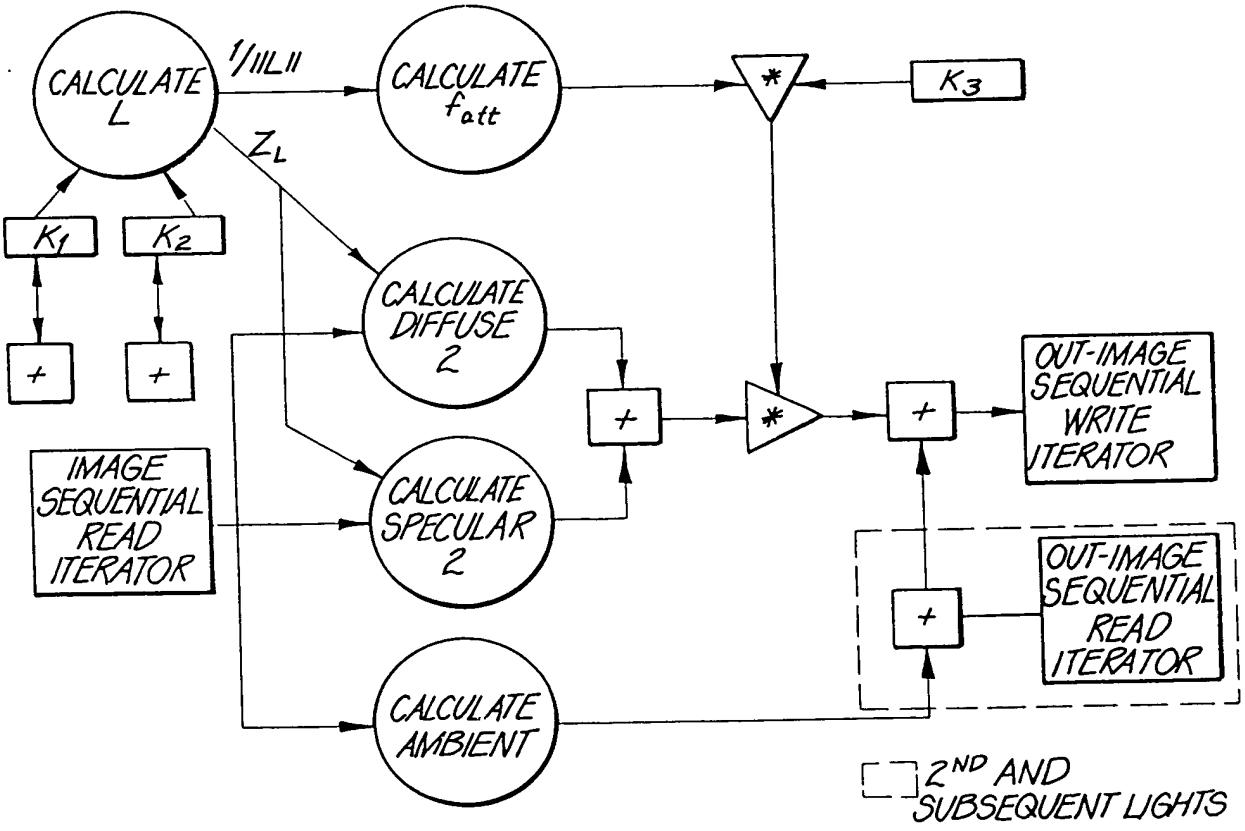

CALCULATE L
1/||L||
CALCULATE f_{att}
K3
Z_L
K1
K2
CALCULATE DIFFUSE 2
OUT-IMAGE SEQUENTIAL WRITE ITERATOR
IMAGE SEQUENTIAL READ ITERATOR
CALCULATE SPECULAR 2
OUT-IMAGE SEQUENTIAL READ ITERATOR
CALCULATE AMBIENT
2ND AND SUBSEQUENT LIGHTS

FIG. 149

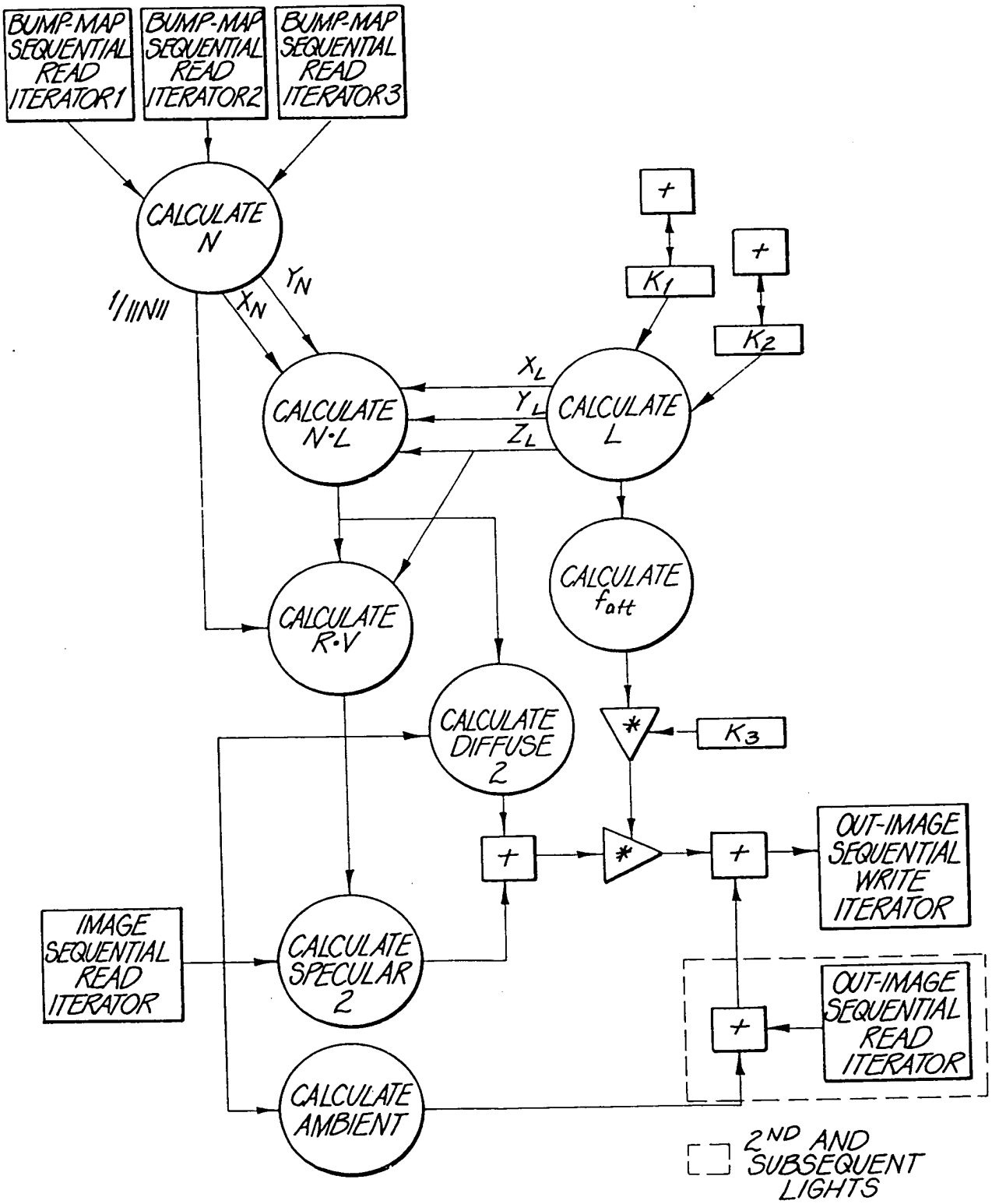

BUMP-MAP SEQUENTIAL READ ITERATOR1
BUMP-MAP SEQUENTIAL READ ITERATOR2
BUMP-MAP SEQUENTIAL READ ITERATOR3
CALCULATE N
1/||N||
X_N
Y_N
+
K_1
+
K_2
CALCULATE N·L
X_L
Y_L
Z_L
CALCULATE L
CALCULATE R·V
CALCULATE f_{att}
CALCULATE DIFFUSE 2
*
K_3
+
*
+
OUT-IMAGE SEQUENTIAL WRITE ITERATOR
IMAGE SEQUENTIAL READ ITERATOR
CALCULATE SPECULAR 2
+
OUT-IMAGE SEQUENTIAL READ ITERATOR
CALCULATE AMBIENT
2ND AND SUBSEQUENT LIGHTS

FIG. 150

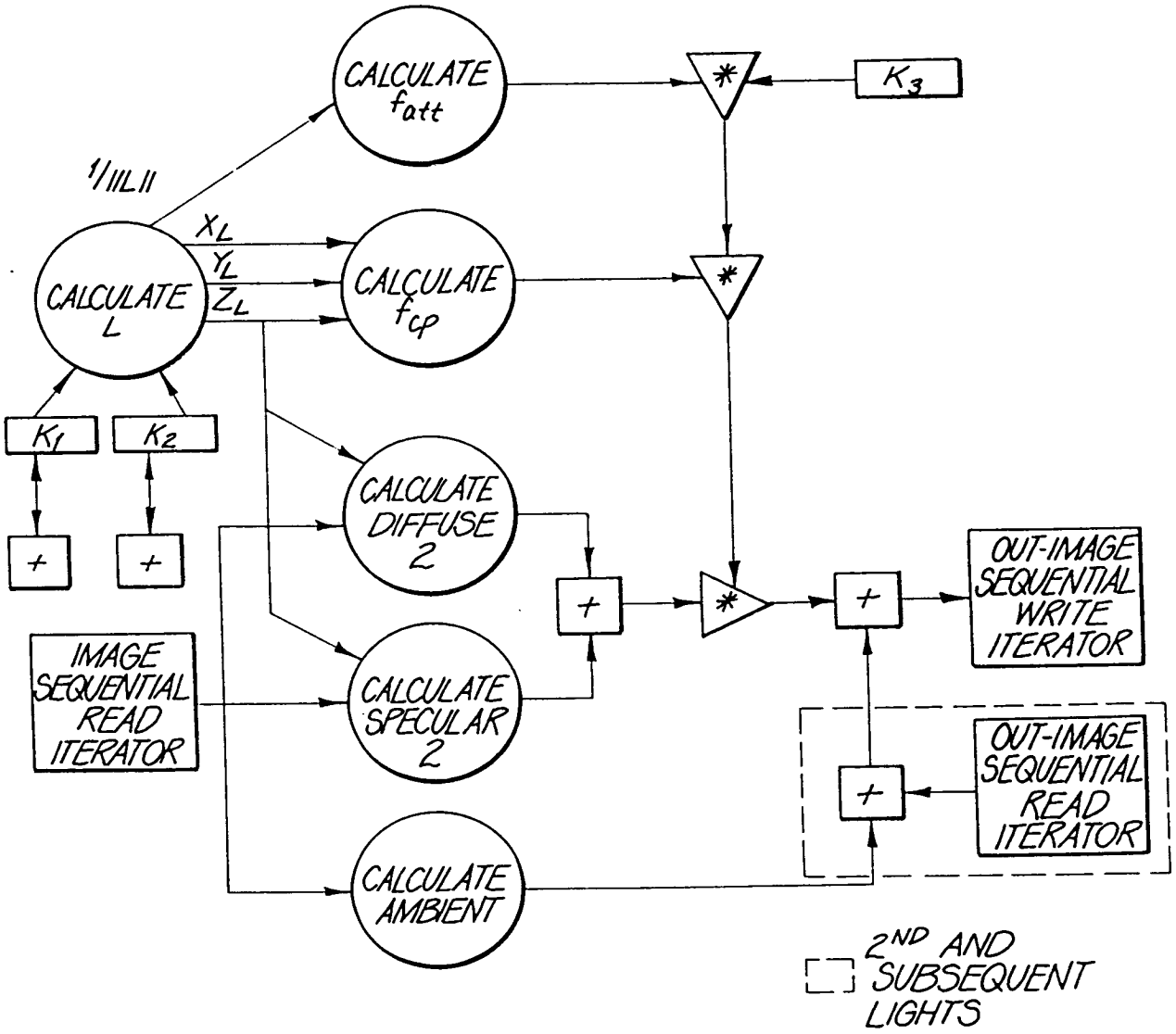

CALCULATE f_{att}
K_3
$1/\|L\|$
X_L
Y_L
Z_L
CALCULATE L
CALCULATE f_{cp}
K_1
K_2
CALCULATE DIFFUSE 2
OUT-IMAGE SEQUENTIAL WRITE ITERATOR
IMAGE SEQUENTIAL READ ITERATOR
CALCULATE SPECULAR 2
OUT-IMAGE SEQUENTIAL READ ITERATOR
CALCULATE AMBIENT
2ND AND SUBSEQUENT LIGHTS

FIG. 151

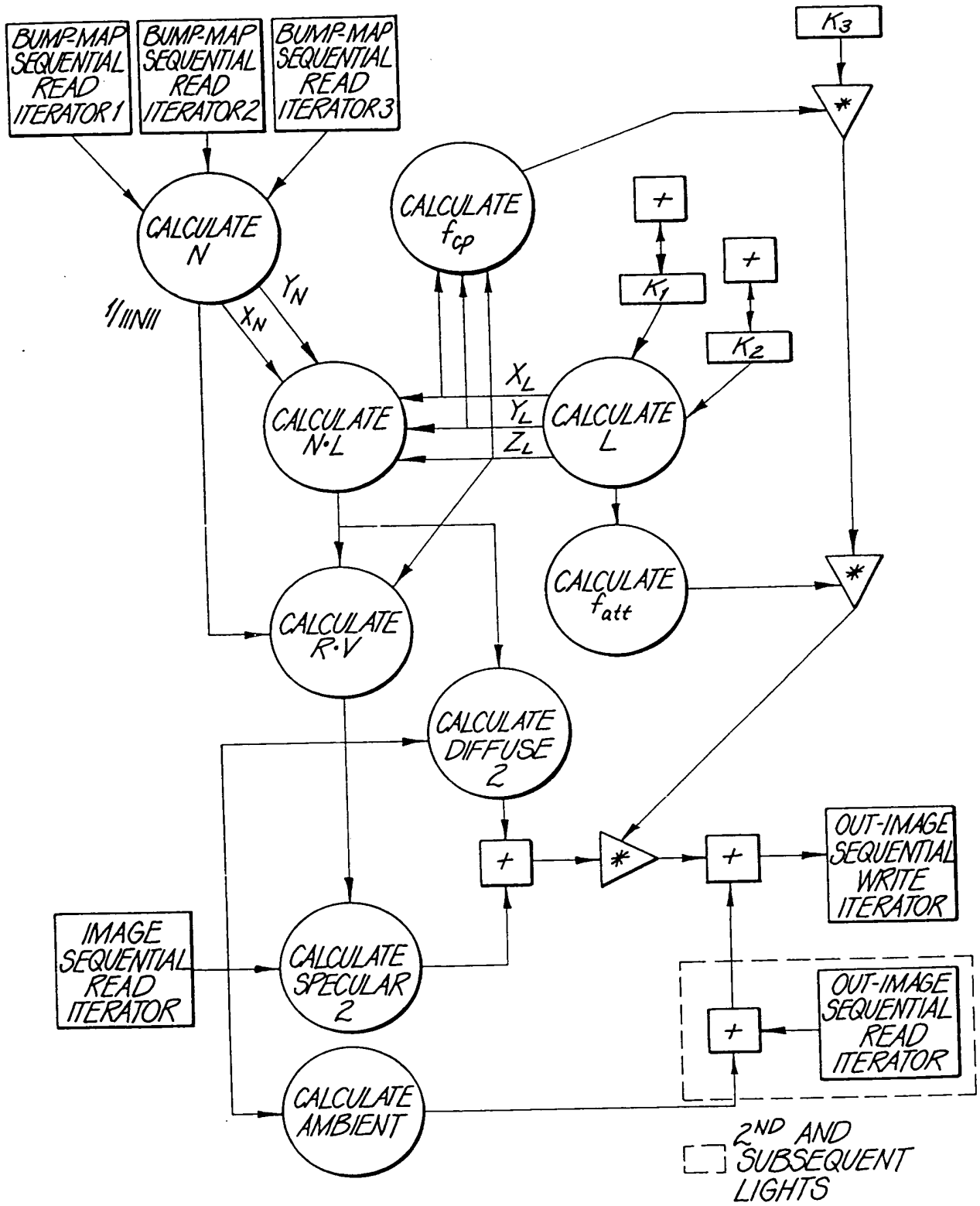

BUMP-MAP SEQUENTIAL READ ITERATOR 1
BUMP-MAP SEQUENTIAL READ ITERATOR 2
BUMP-MAP SEQUENTIAL READ ITERATOR 3
CALCULATE N
1/||N||
Y_N
X_N
CALCULATE f_{cp}
K_3
*
+
K_1
+
K_2
X_L
Y_L
Z_L
CALCULATE N·L
CALCULATE L
CALCULATE f_{att}
CALCULATE R·V
CALCULATE DIFFUSE 2
+
*
+
OUT-IMAGE SEQUENTIAL WRITE ITERATOR
IMAGE SEQUENTIAL READ ITERATOR
CALCULATE SPECULAR 2
+
OUT-IMAGE SEQUENTIAL READ ITERATOR
CALCULATE AMBIENT
2ND AND SUBSEQUENT LIGHTS

FIG. 152

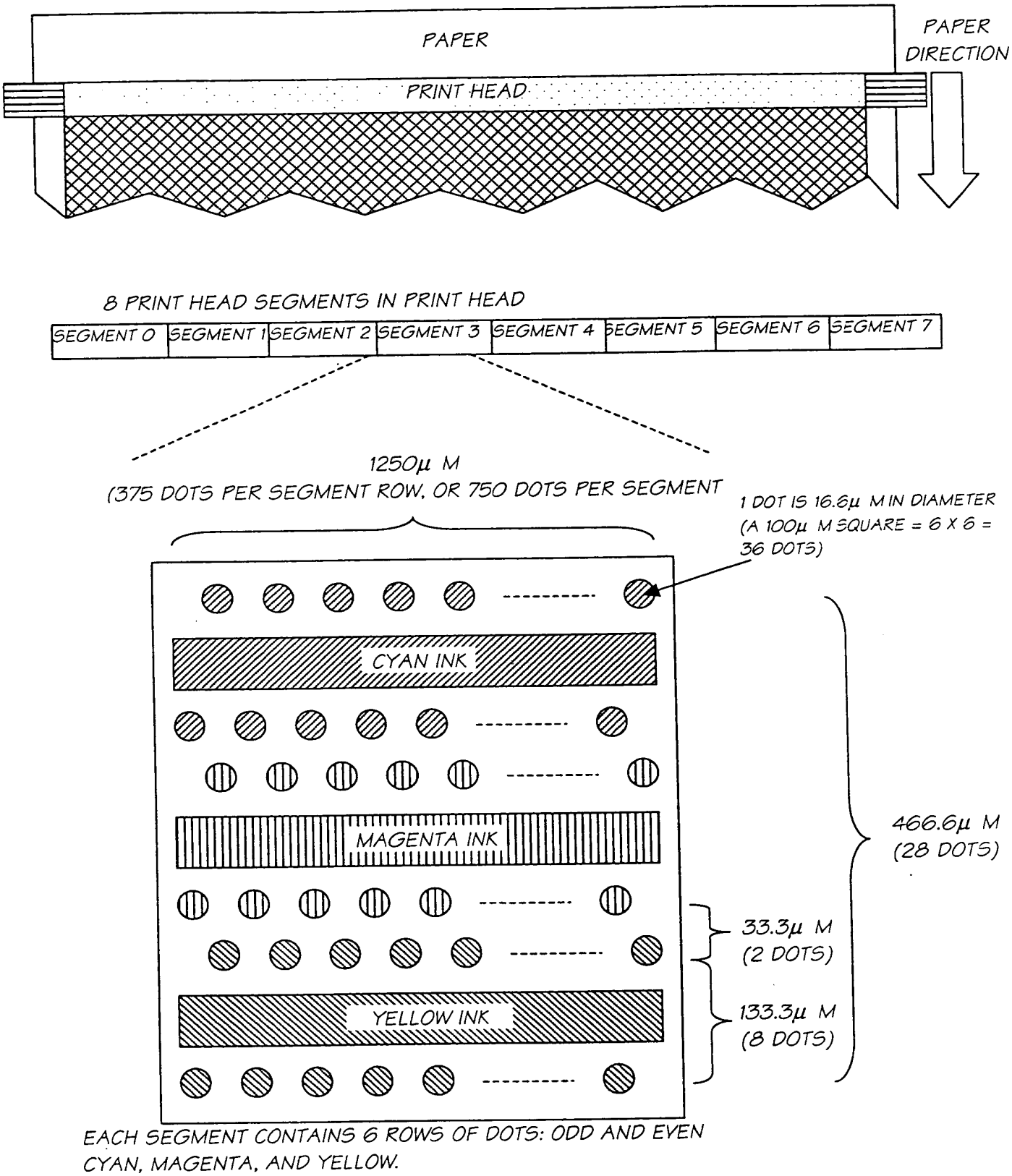

PAPER
PAPER DIRECTION
PRINT HEAD
8 PRINT HEAD SEGMENTS IN PRINT HEAD
SEGMENT 0
SEGMENT 1
SEGMENT 2
SEGMENT 3
SEGMENT 4
SEGMENT 5
SEGMENT 6
SEGMENT 7
1250μ M
(375 DOTS PER SEGMENT ROW, OR 750 DOTS PER SEGMENT
1 DOT IS 16.6μ M IN DIAMETER
(A 100μ M SQUARE = 6 X 6 = 36 DOTS)
CYAN INK
MAGENTA INK
YELLOW INK
466.6μ M
(28 DOTS)
33.3μ M
(2 DOTS)
133.3μ M
(8 DOTS)
EACH SEGMENT CONTAINS 6 ROWS OF DOTS: ODD AND EVEN
CYAN, MAGENTA, AND YELLOW.

FIG. 153

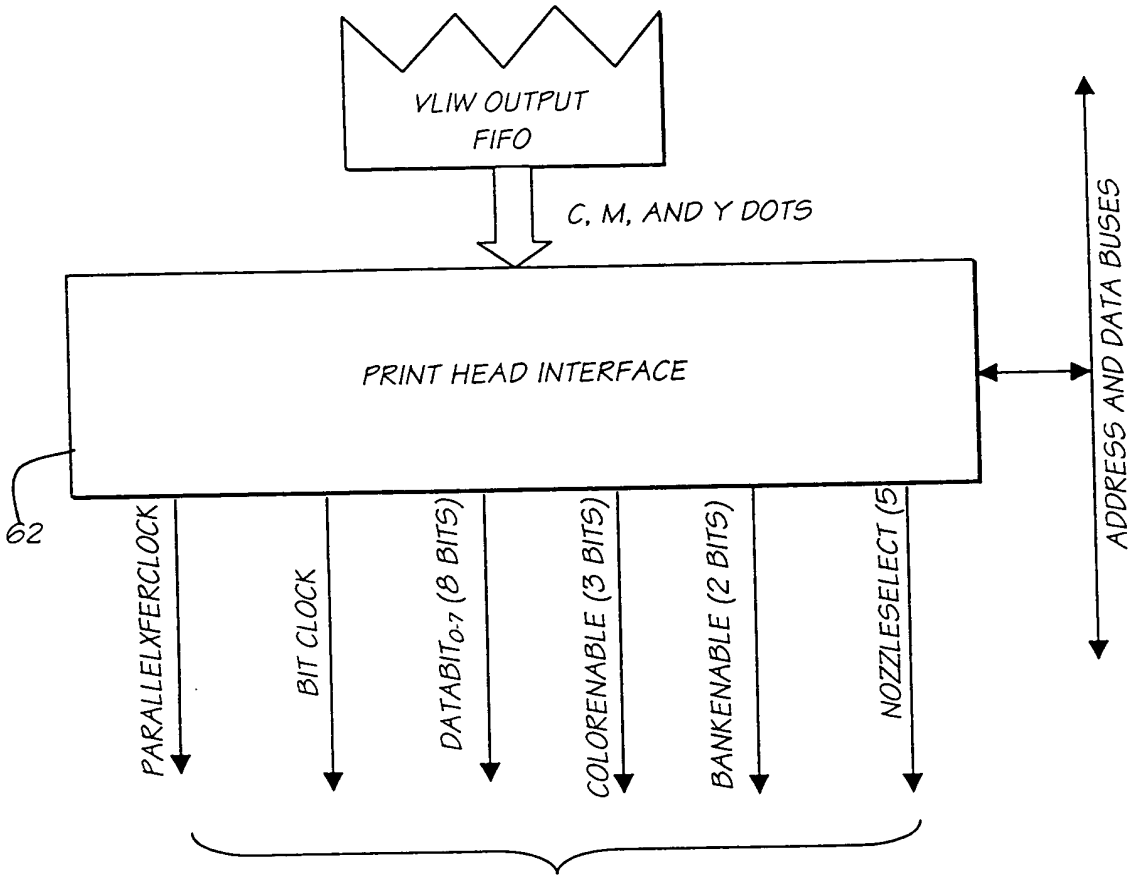


FIG. 154

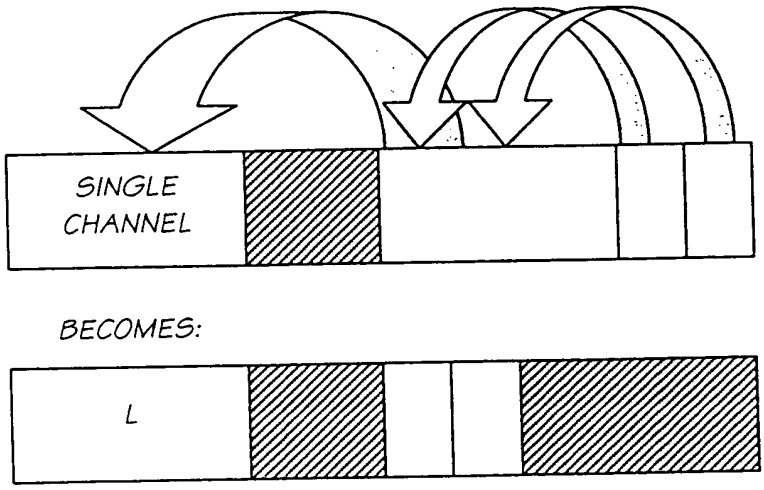


FIG. 155

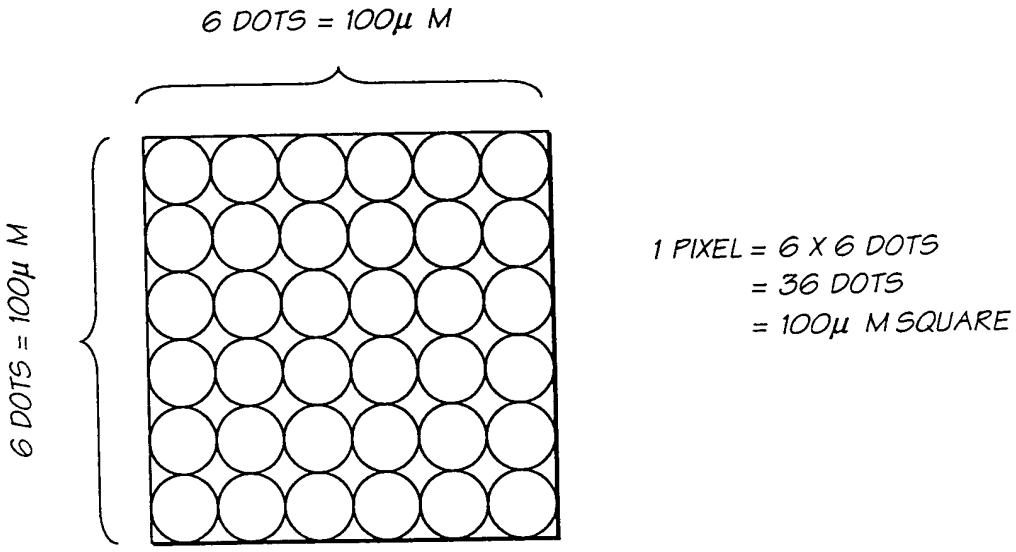

6 DOTS = 100μ M
6 DOTS = 100μ M
1 PIXEL = 6 X 6 DOTS
= 36 DOTS
= 100μ M SQUARE

FIG. 156

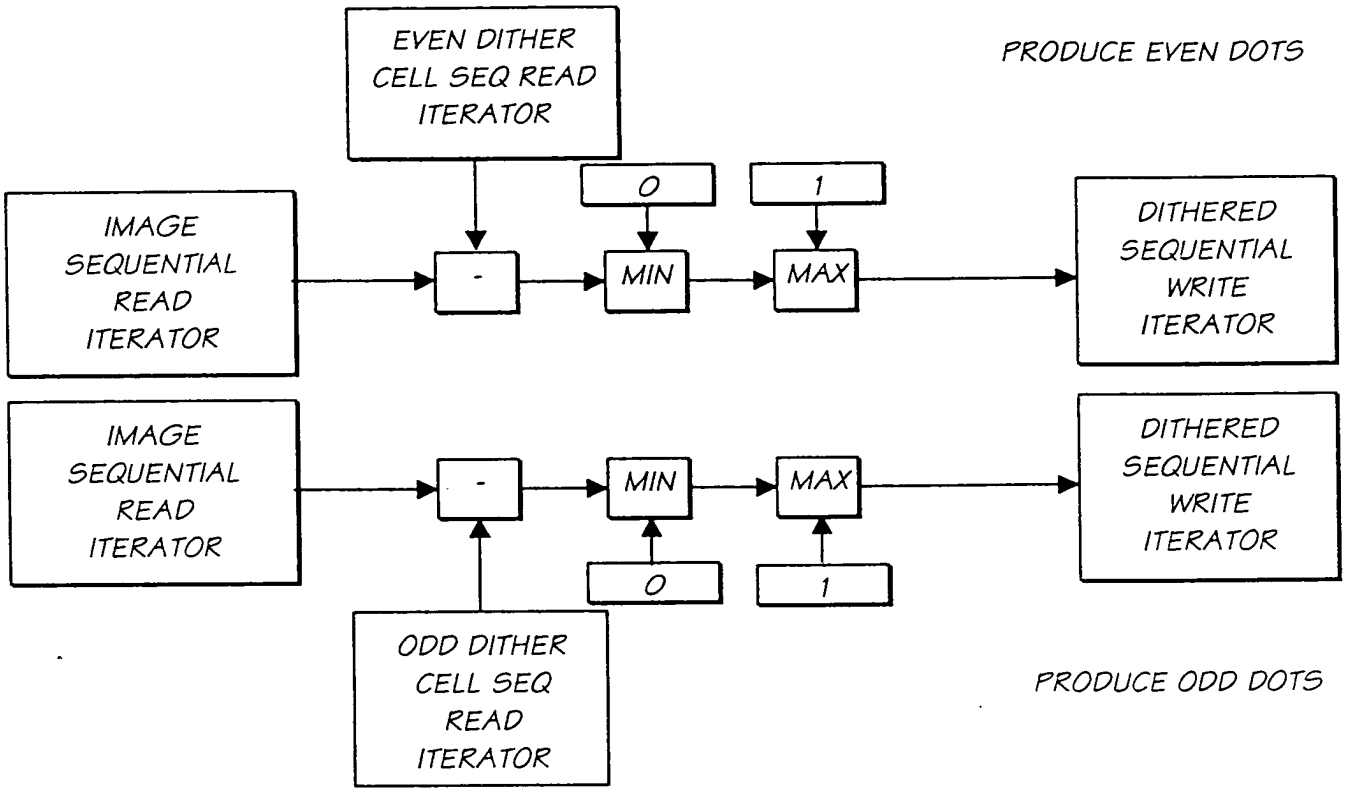


FIG. 157

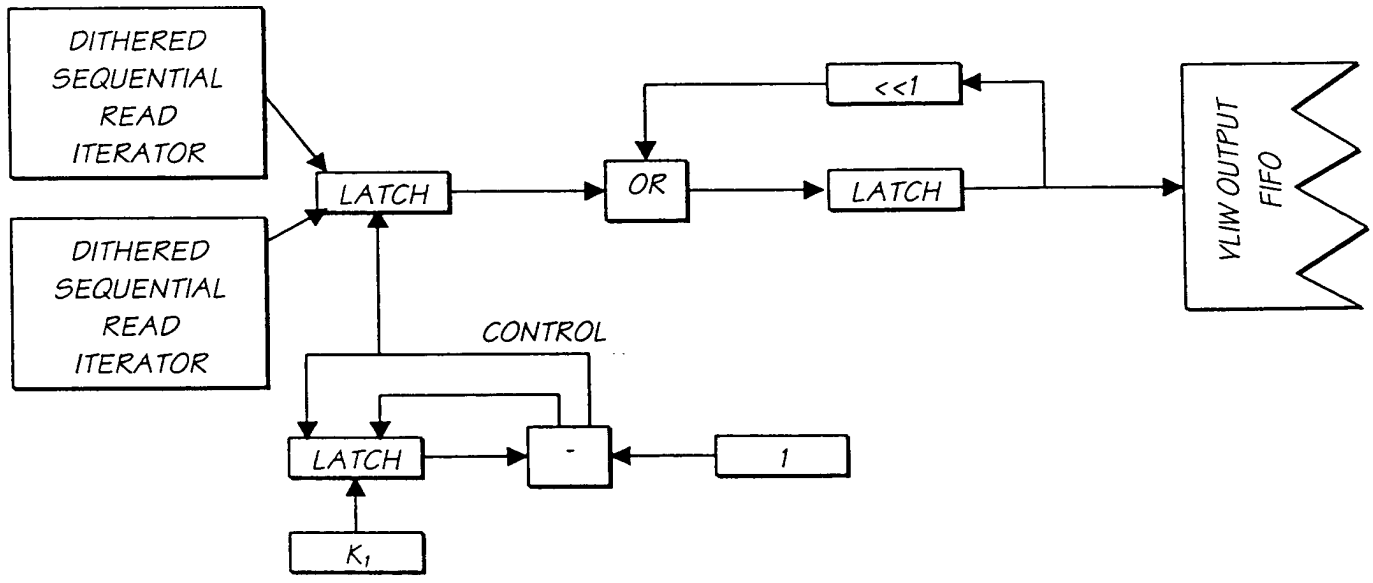


FIG. 158

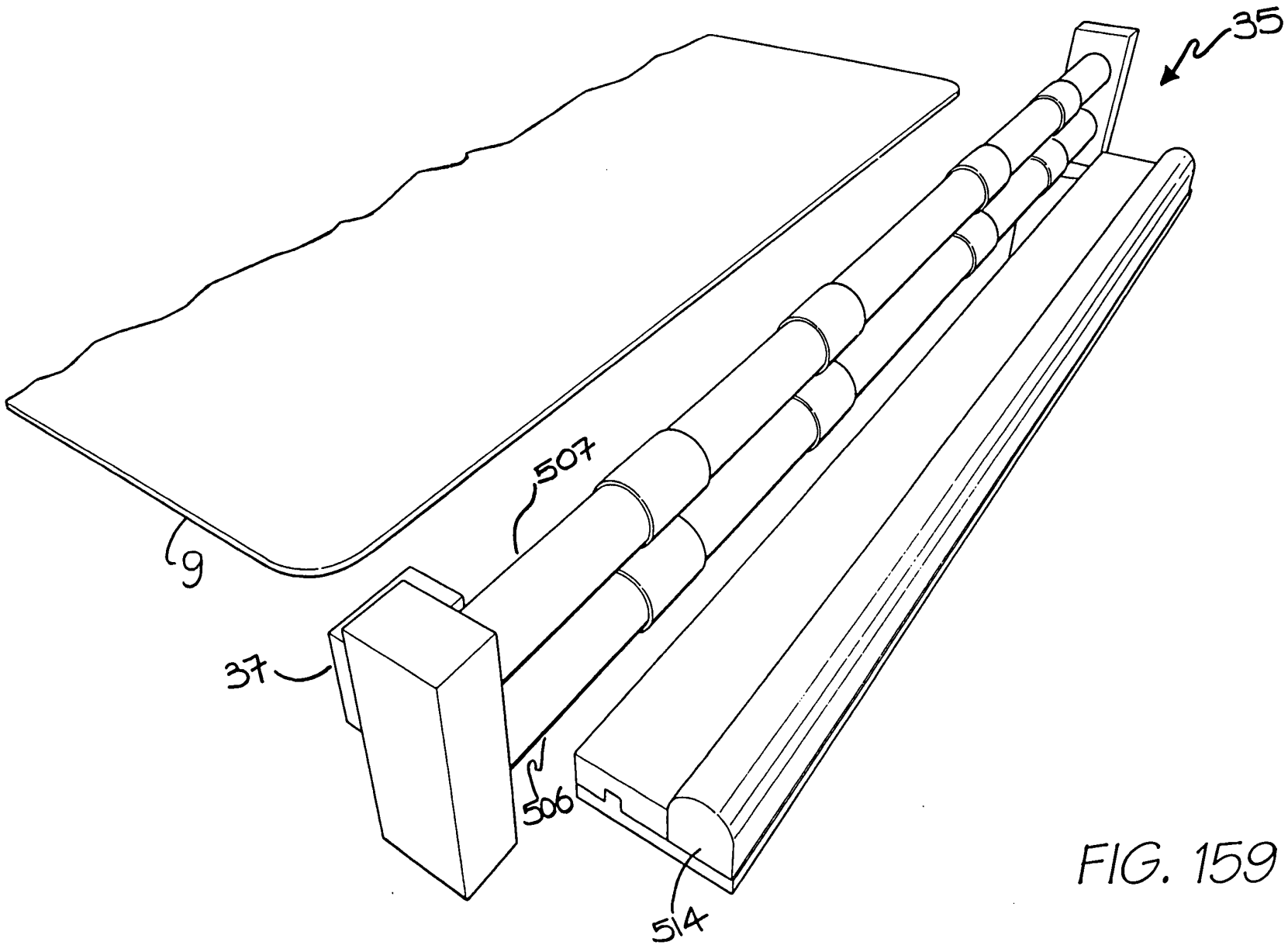


FIG. 159

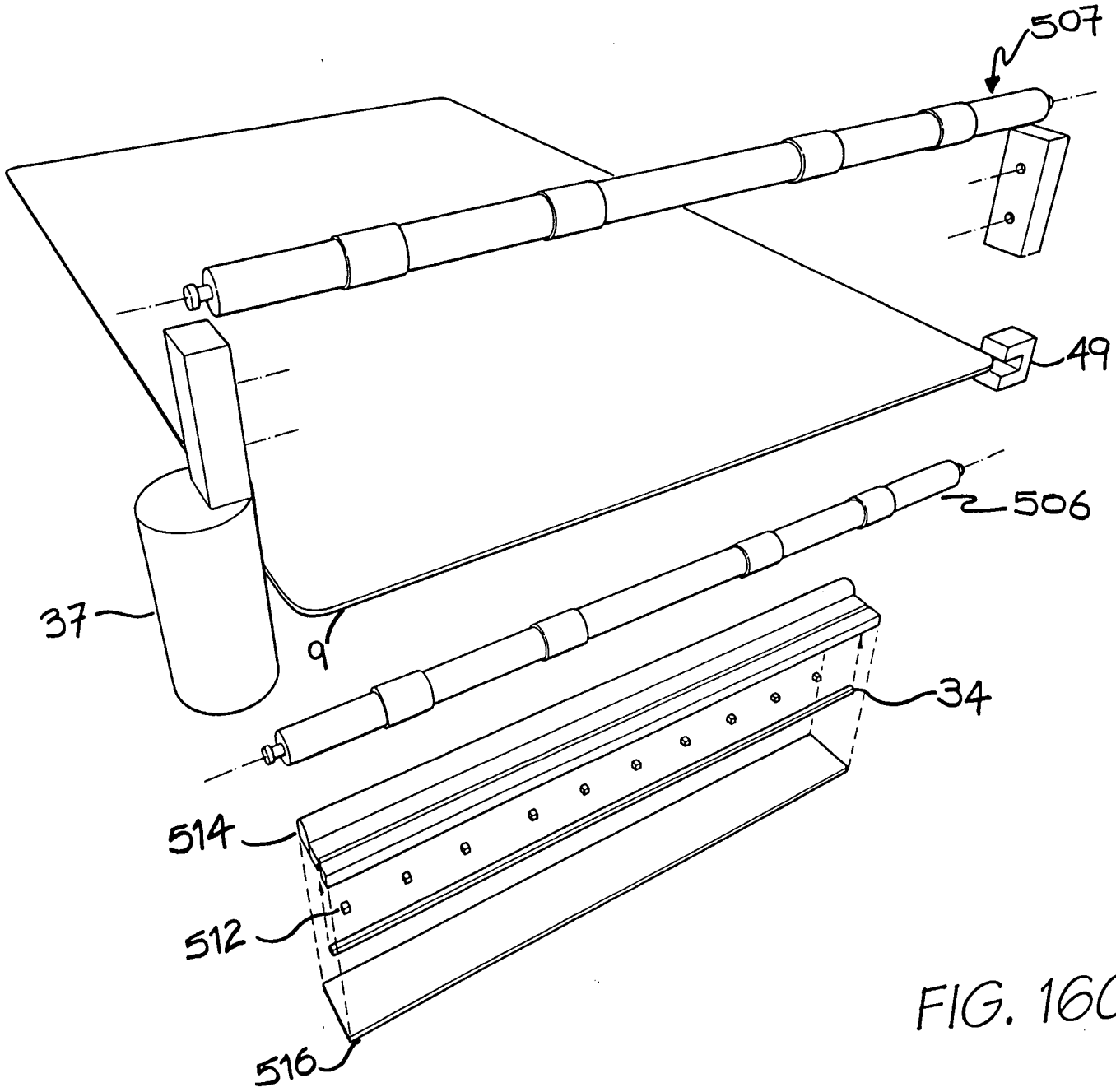


FIG. 160

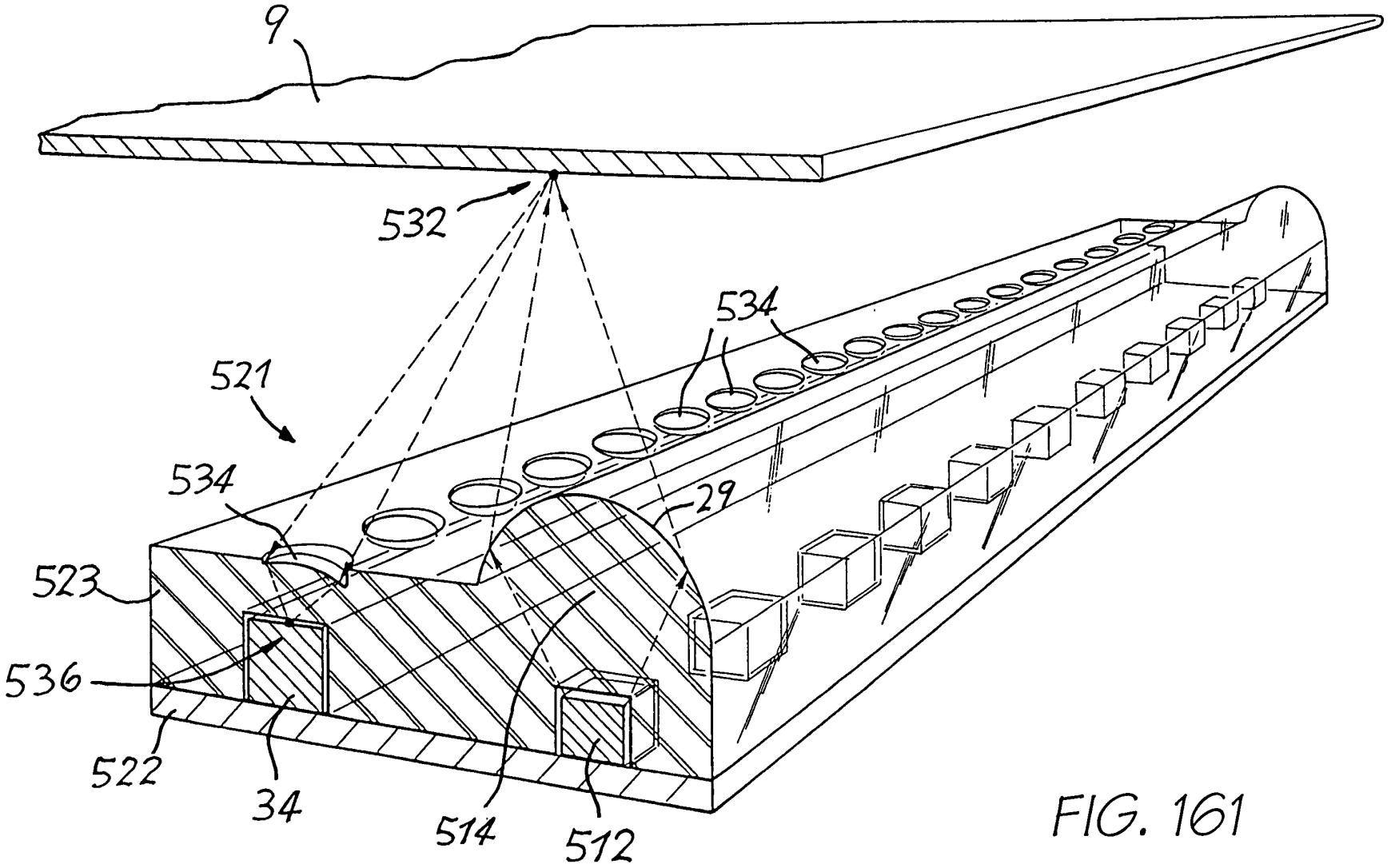


FIG. 161

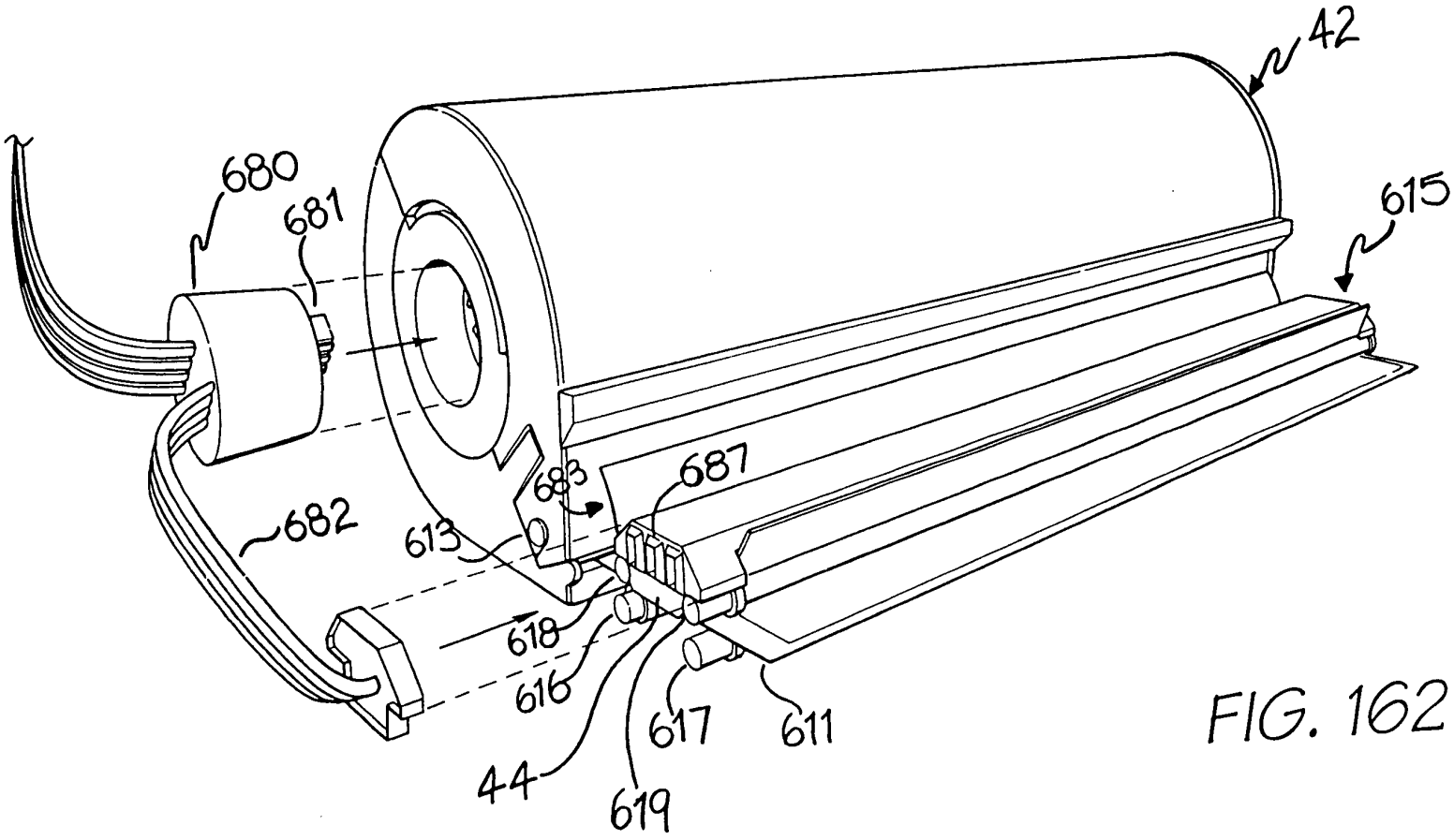


FIG. 162

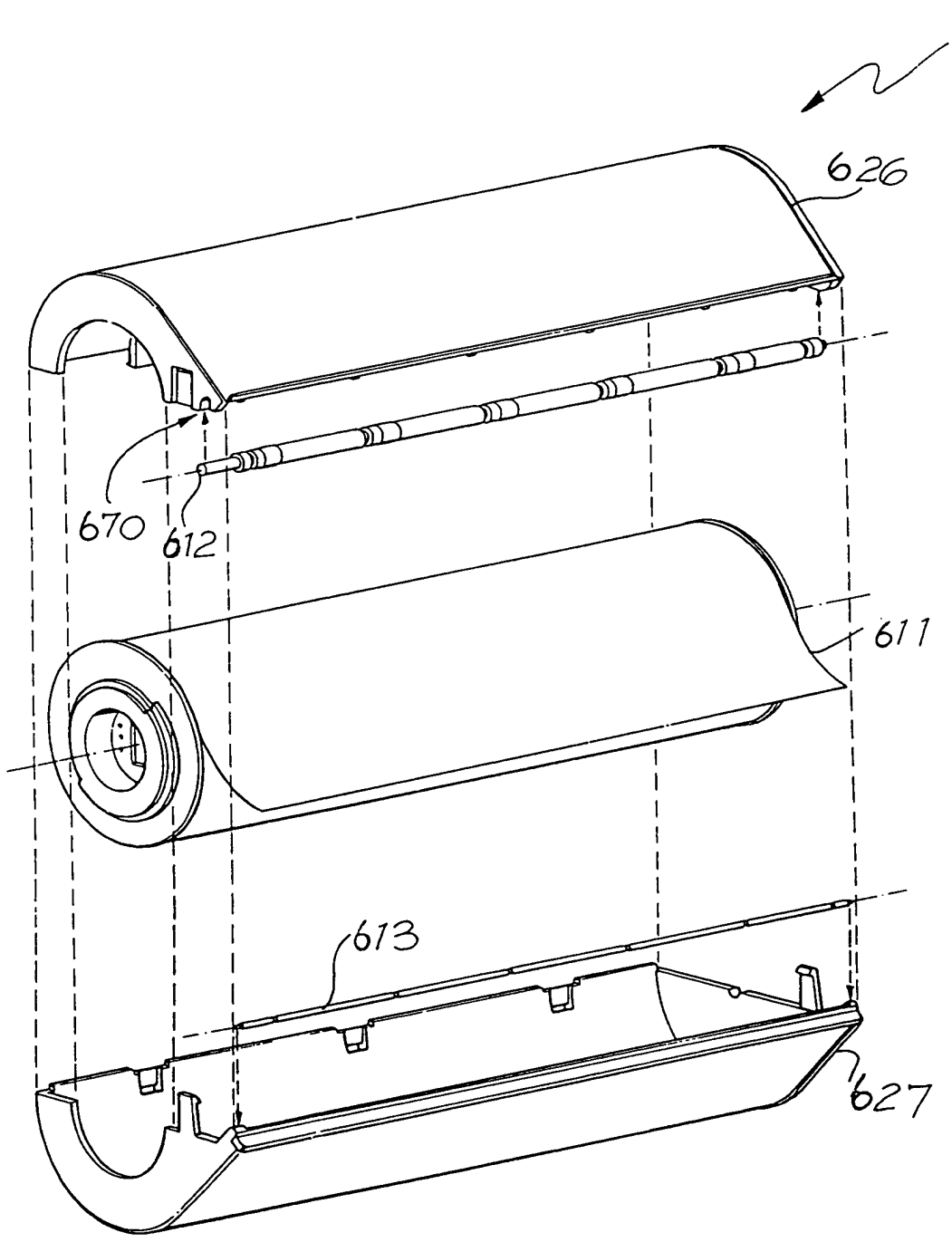


FIG. 163

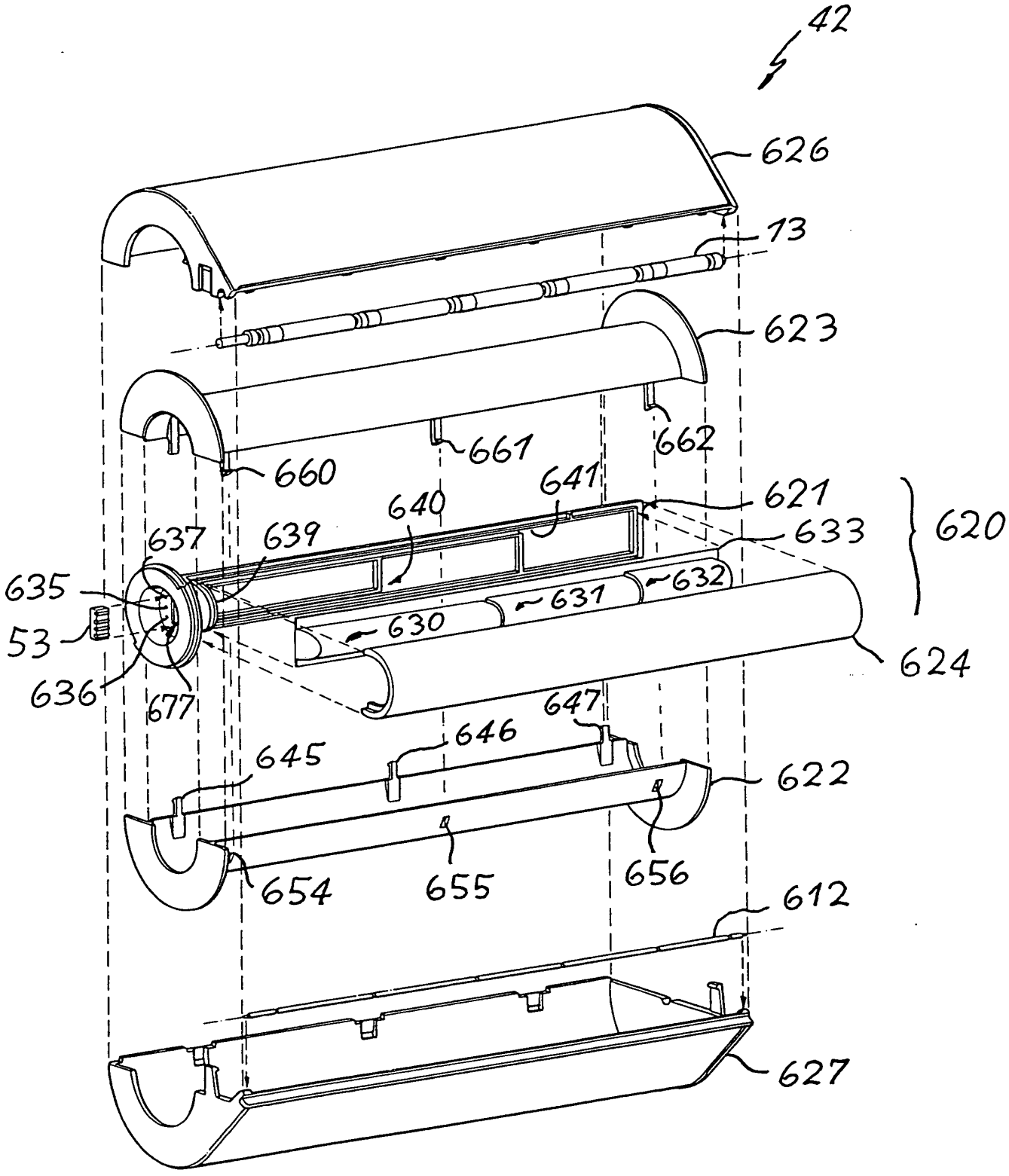

42
626
73
623
660
661
662
641
640
621
620
633
637
639
635
632
631
630
53
624
636
677
647
646
645
622
654
655
656
612
627

FIG. 164

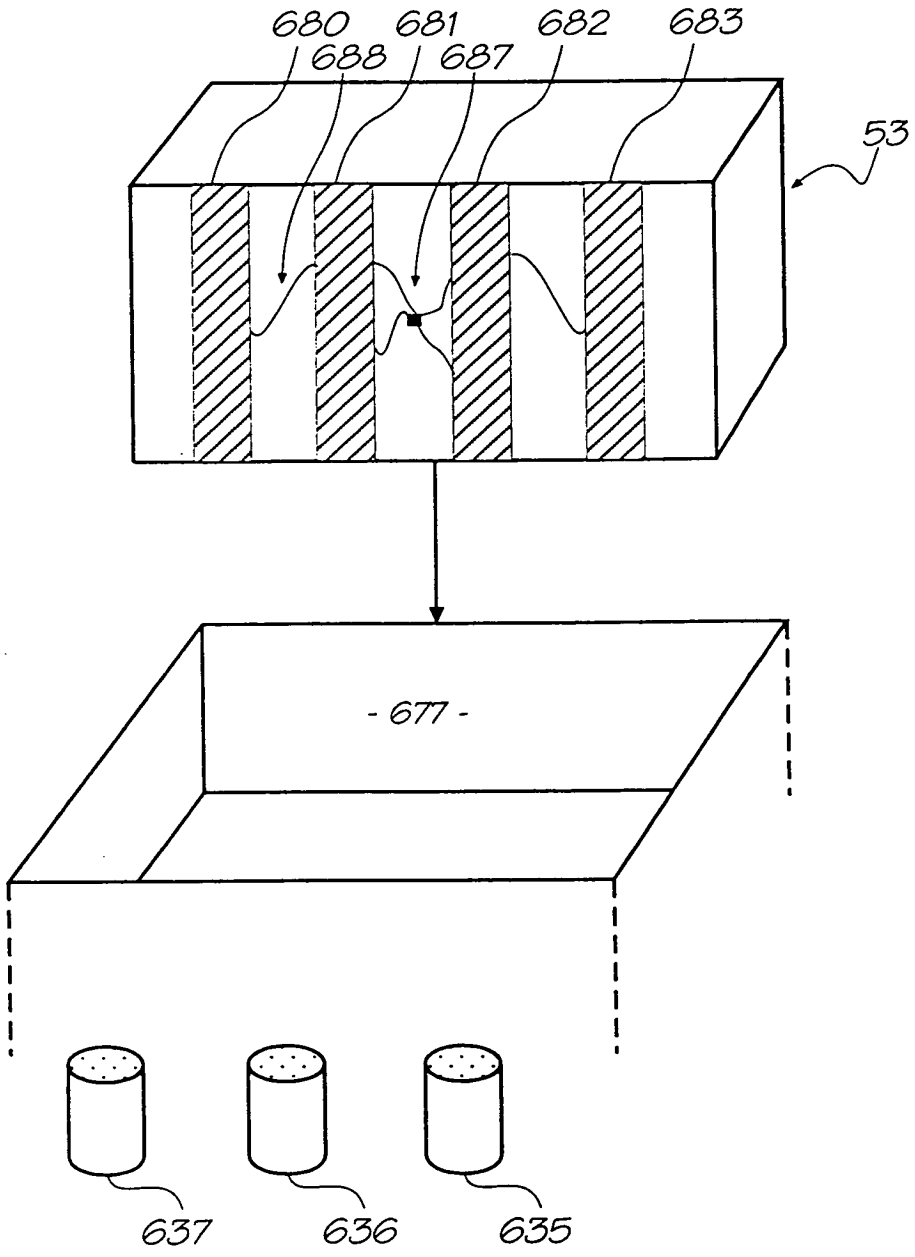


FIG. 165

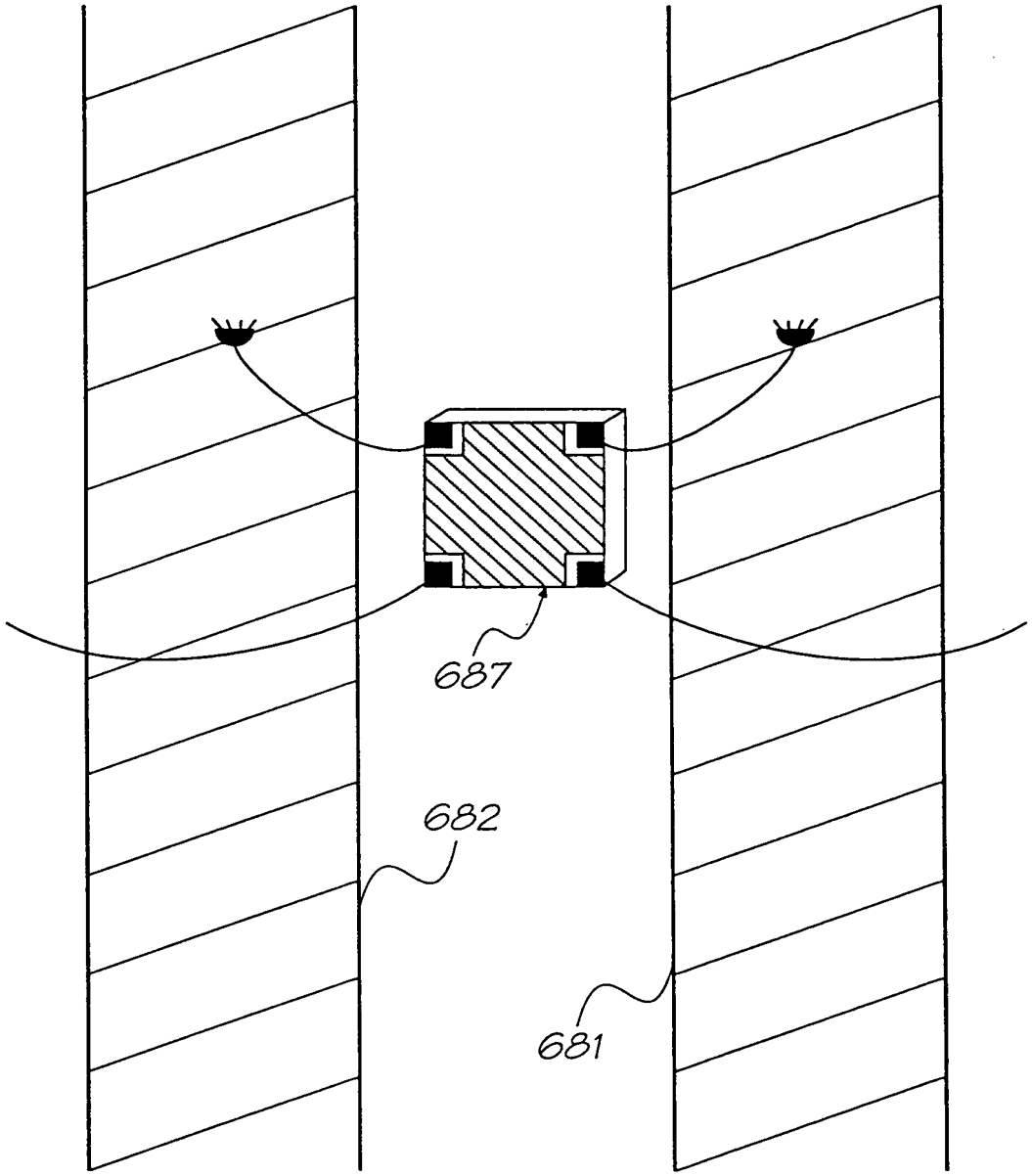


FIG. 166

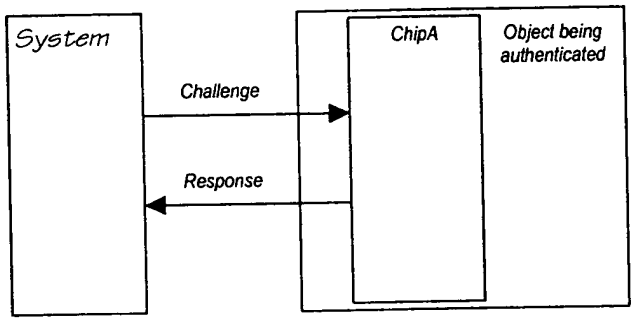


FIG. 167

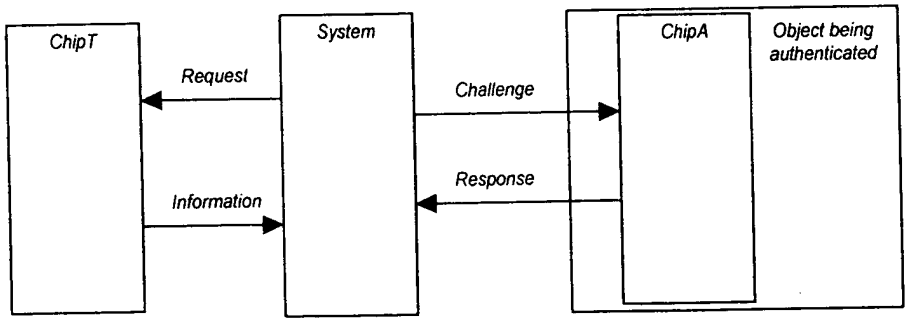


FIG. 168

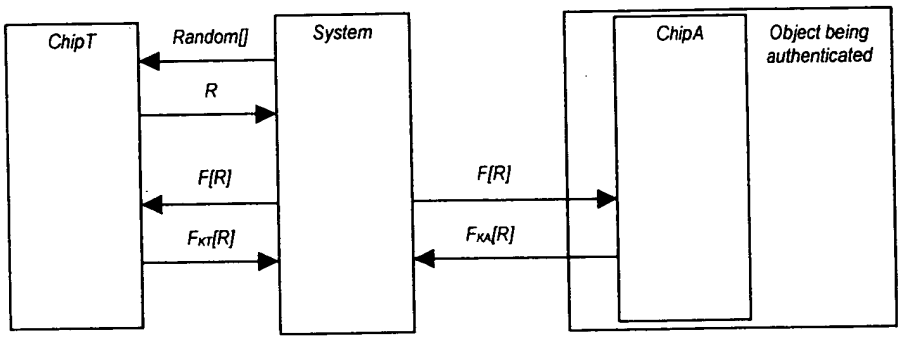


FIG. 169

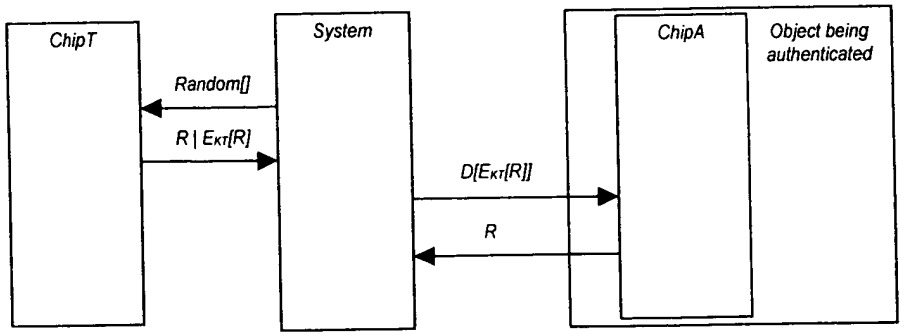


FIG. 170

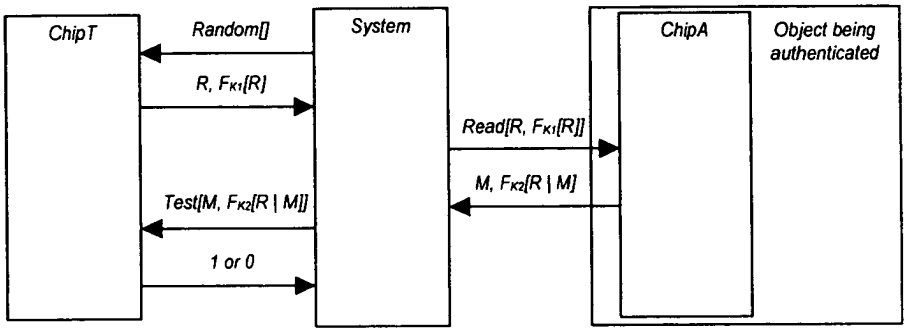


FIG. 171

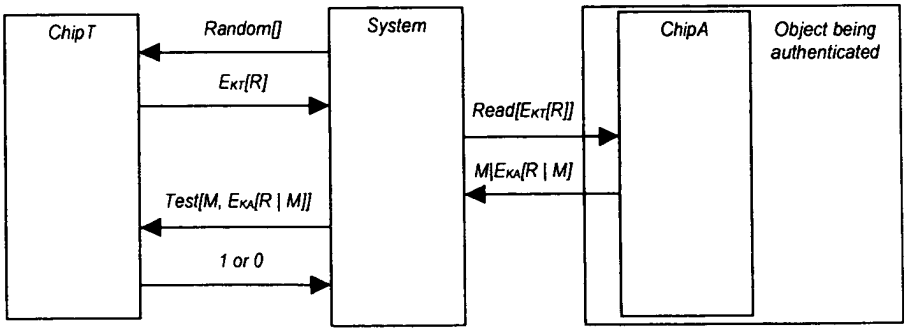


FIG. 172

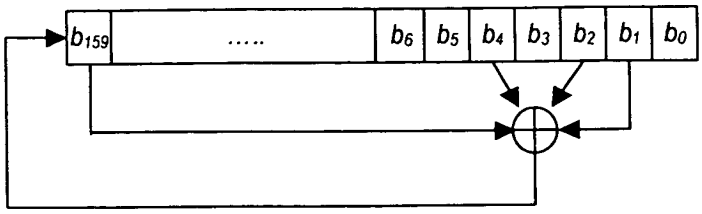


FIG. 173

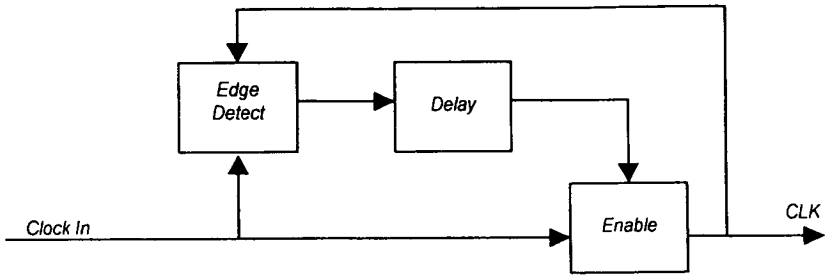


FIG. 174

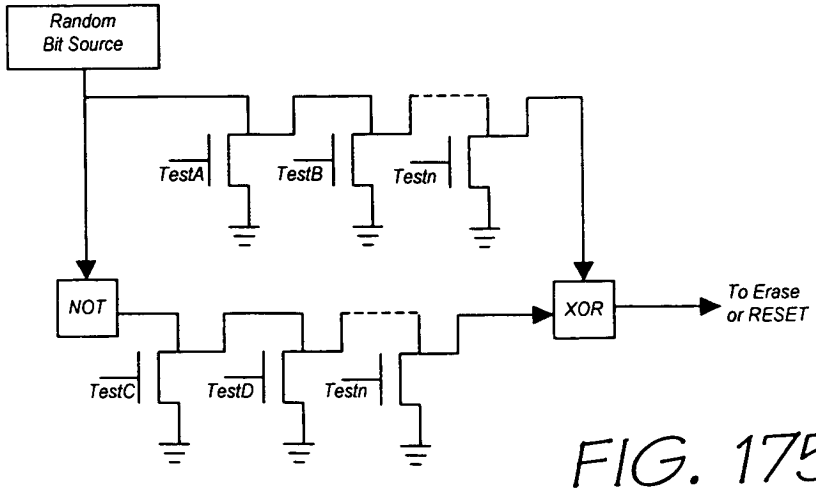


FIG. 175

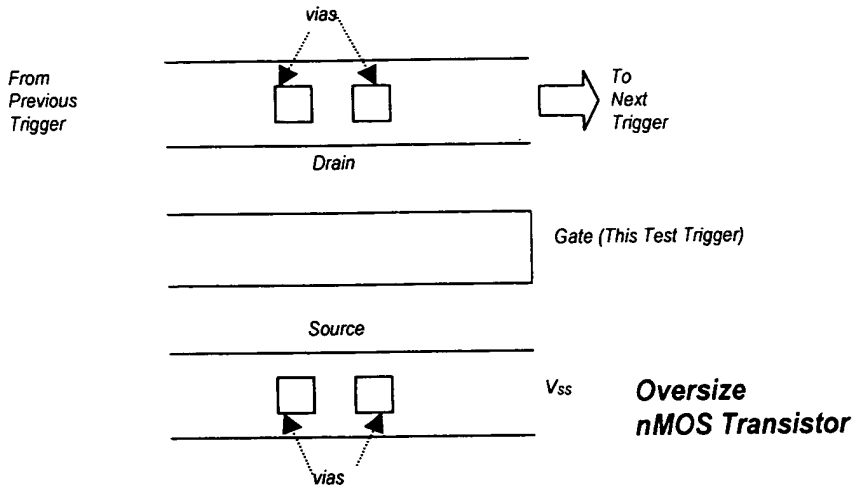


FIG. 176

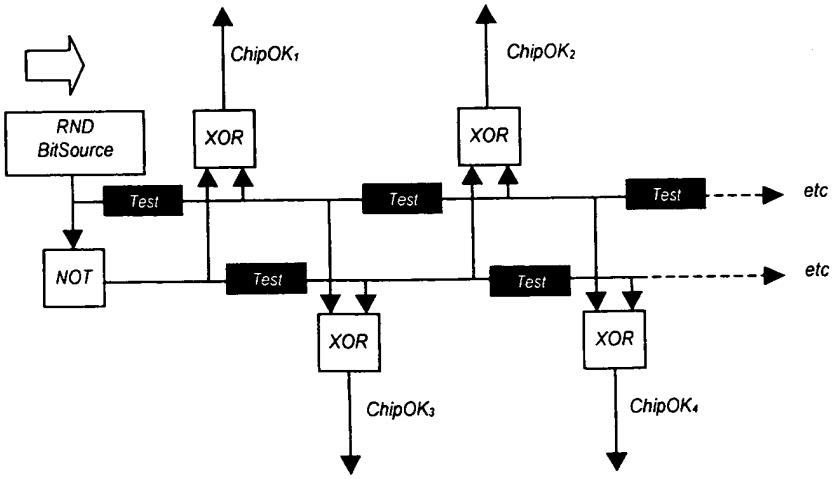


FIG. 177

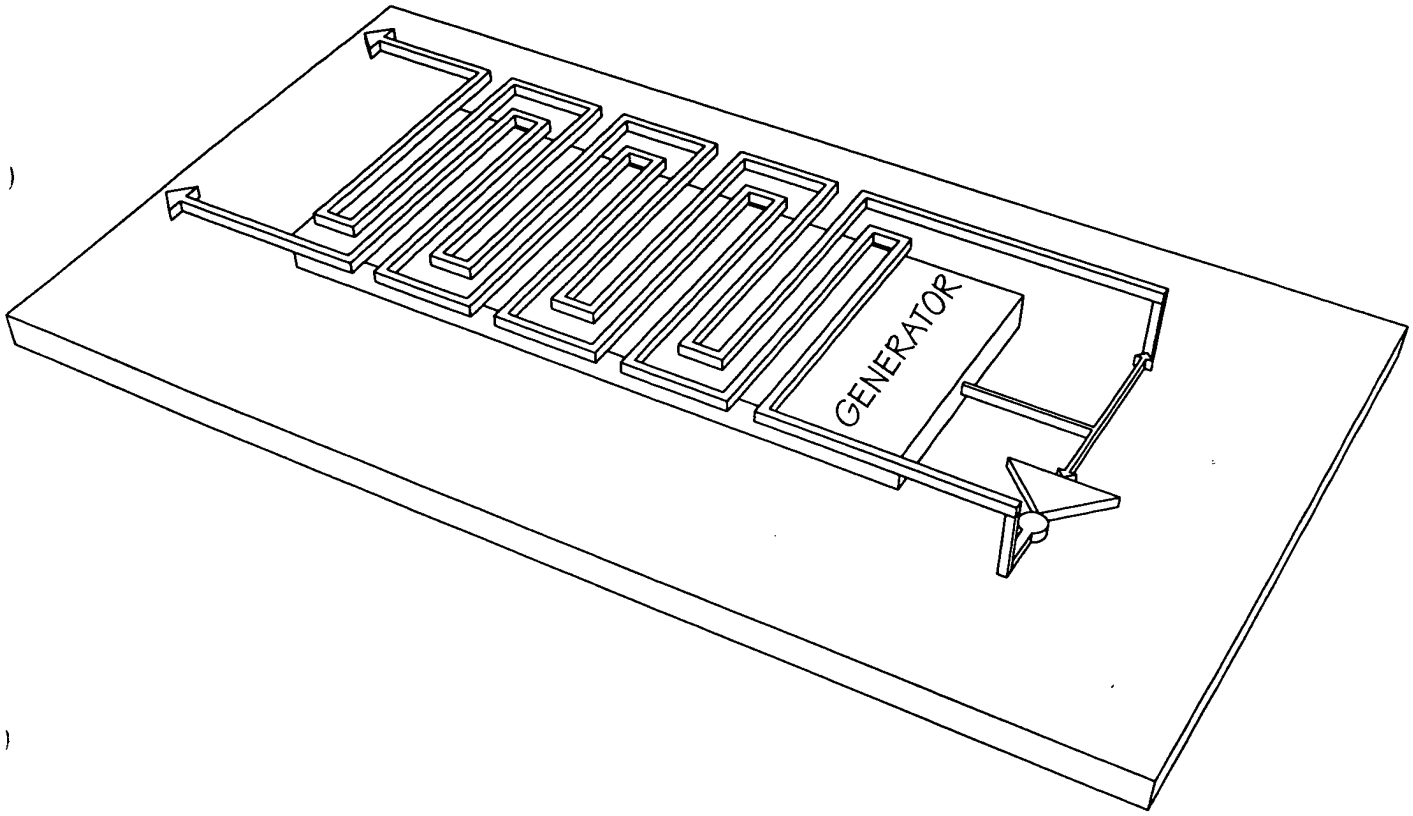


FIG. 178

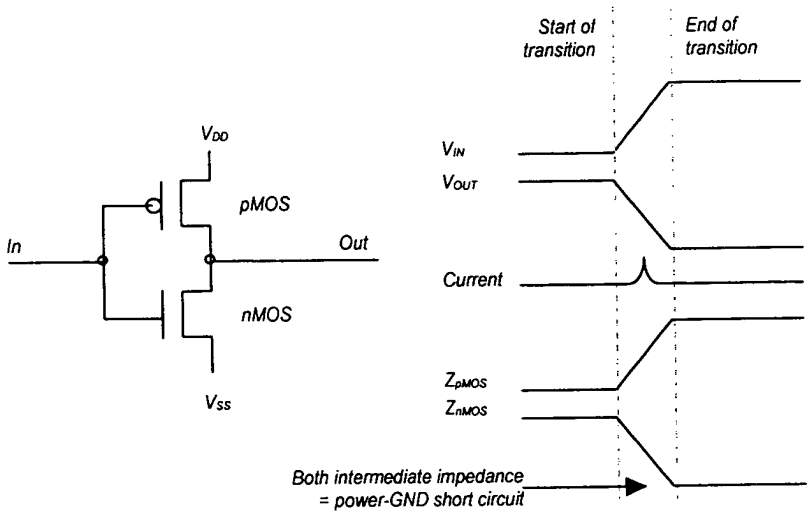


FIG. 179

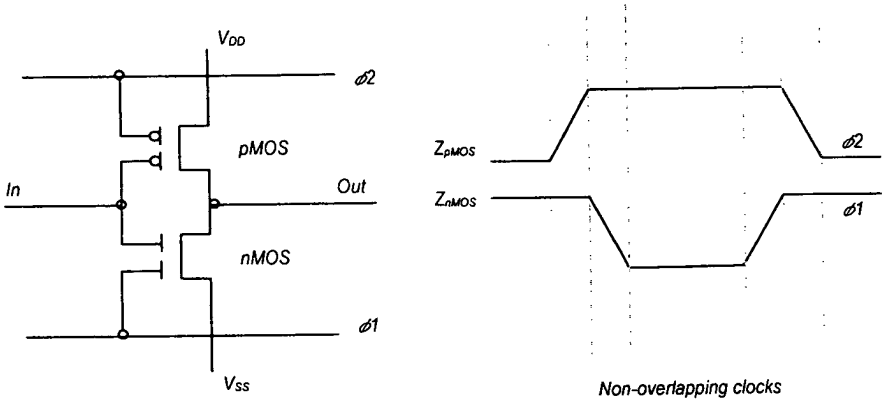


FIG. 180

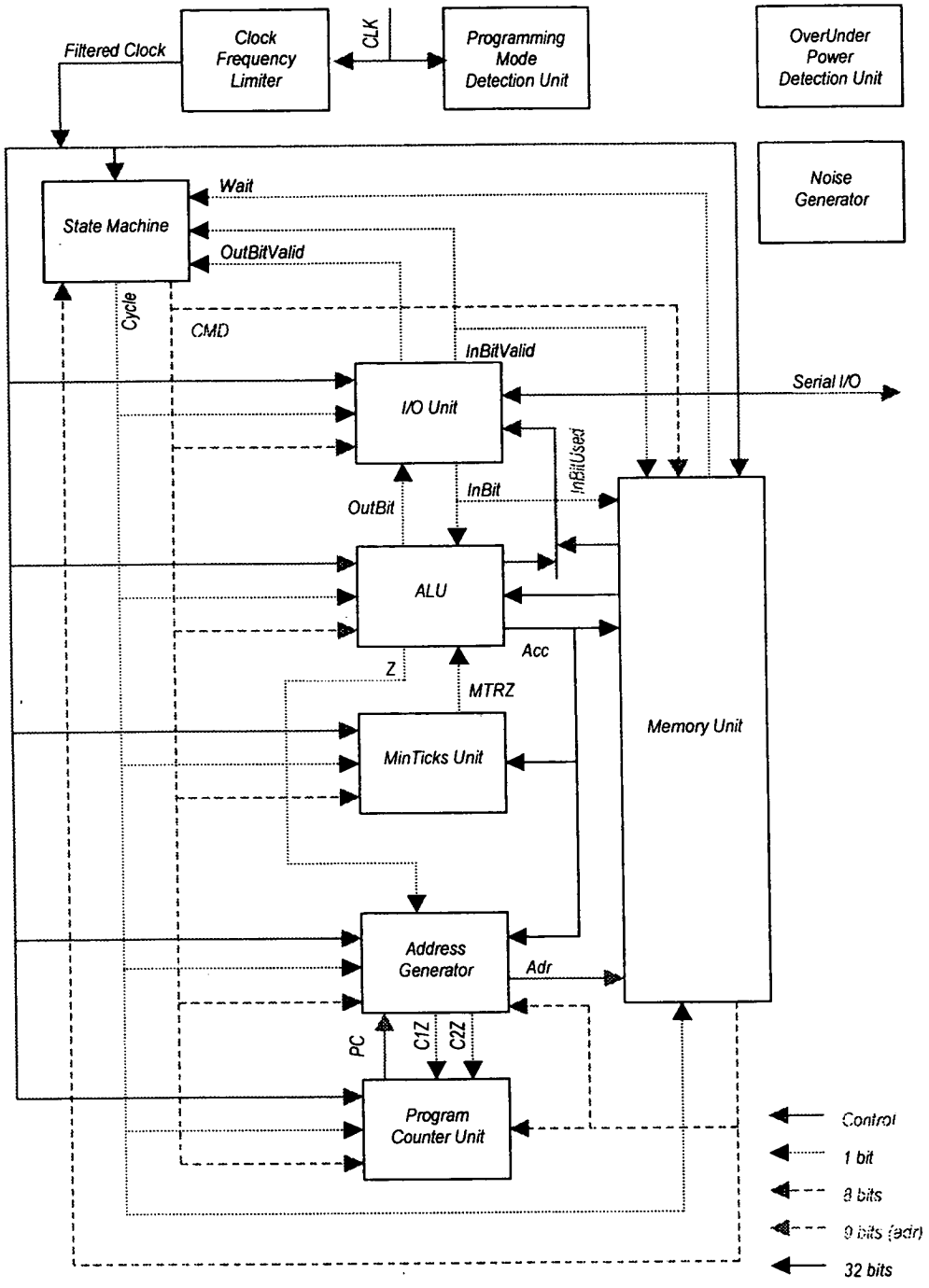

Filtered Clock
Clock Frequency Limiter
CLK
Programming Mode Detection Unit
OverUnder Power Detection Unit
Noise Generator
State Machine
Wait
OutBitValid
Cycle
CMD
InBitValid
I/O Unit
Serial I/O
InBitUsed
InBit
OutBit
ALU
Acc
Z
MTRZ
MinTicks Unit
Memory Unit
Address Generator
Adr
PC
C1Z
C2Z
Program Counter Unit
Control
1 bit
8 bits
9 bits (adr)
32 bits

FIG. 181

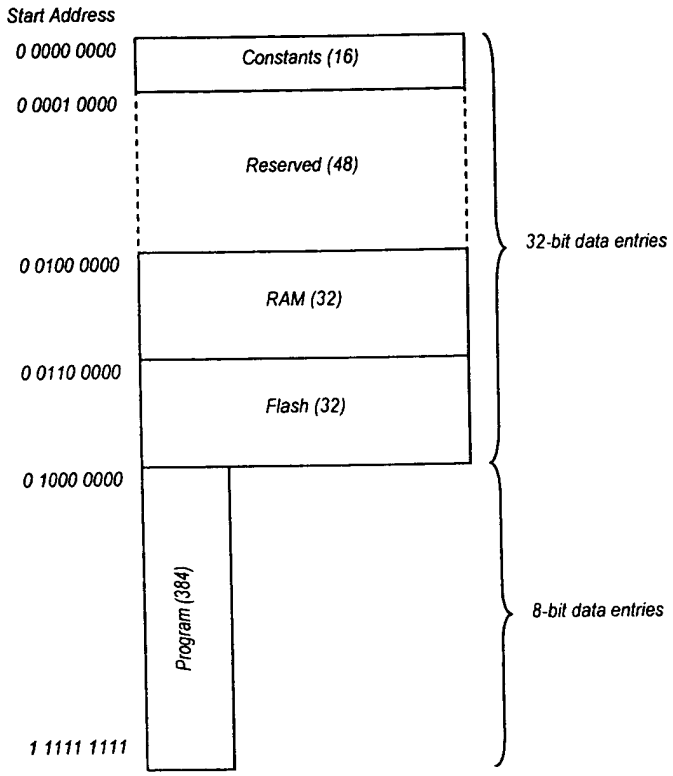

Start Address
0 0000 0000
Constants (16)
0 0001 0000
Reserved (48)
0 0100 0000
RAM (32)
0 0110 0000
Flash (32)
0 1000 0000
Program (384)
1 1111 1111
32-bit data entries
8-bit data entries

FIG. 182

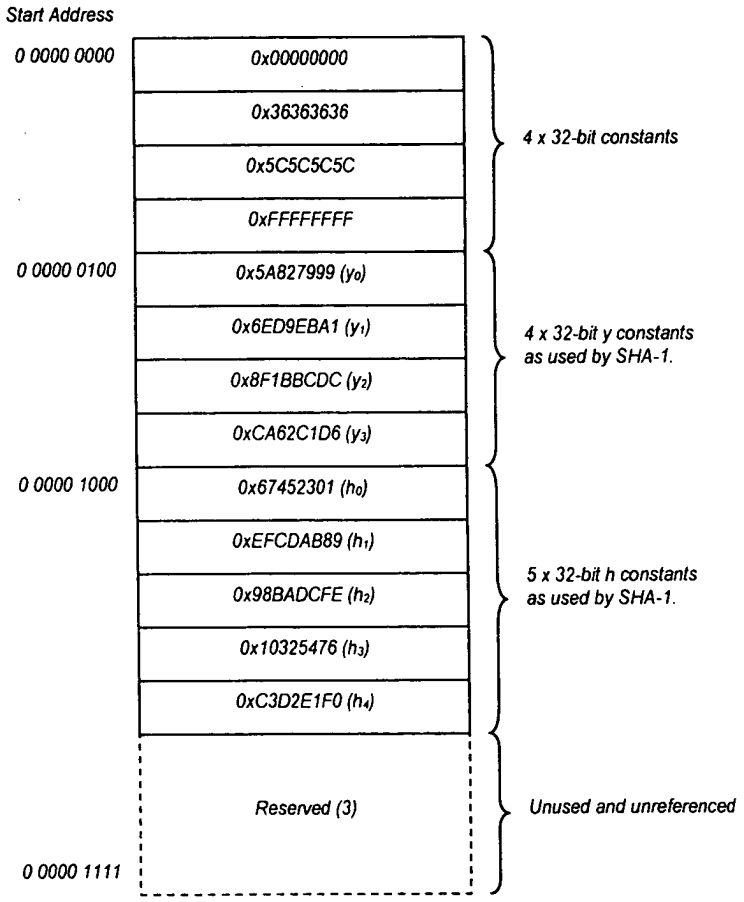

Start Address
0 0000 0000
0x00000000
0x36363636
0x5C5C5C5C
0xFFFFFFFF
4 x 32-bit constants
0 0000 0100
0x5A827999 (y0)
0x6ED9EBA1 (y1)
0x8F1BBCDC (y2)
0xCA62C1D6 (y3)
4 x 32-bit y constants as used by SHA-1.
0 0000 1000
0x67452301 (h0)
0xEFCDAB89 (h1)
0x98BADCFE (h2)
0x10325476 (h3)
0xC3D2E1F0 (h4)
5 x 32-bit h constants as used by SHA-1.
Reserved (3)
Unused and unreferenced
0 0000 1111

FIG. 183

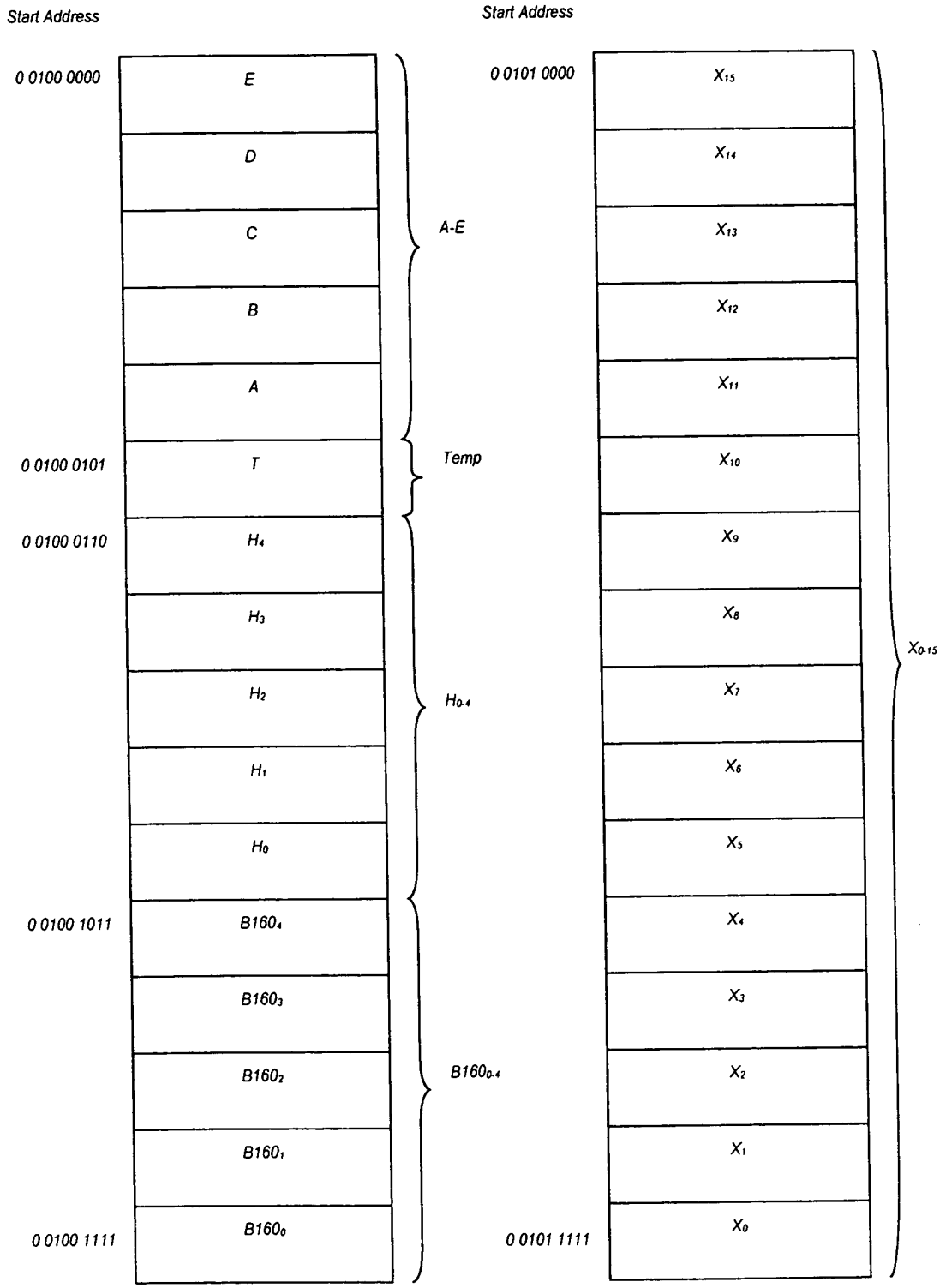

Start Address
0 0100 0000
E
D
C
B
A
A-E
0 0100 0101
T
Temp
0 0100 0110
H4
H3
H2
H1
H0
H0-4
0 0100 1011
B160 4
B160 3
B160 2
B160 1
B160 0
B160 0-4
0 0100 1111
Start Address
0 0101 0000
X15
X14
X13
X12
X11
X10
X9
X8
X7
X6
X5
X4
X3
X2
X1
X0
X0-15
0 0101 1111

FIG. 184

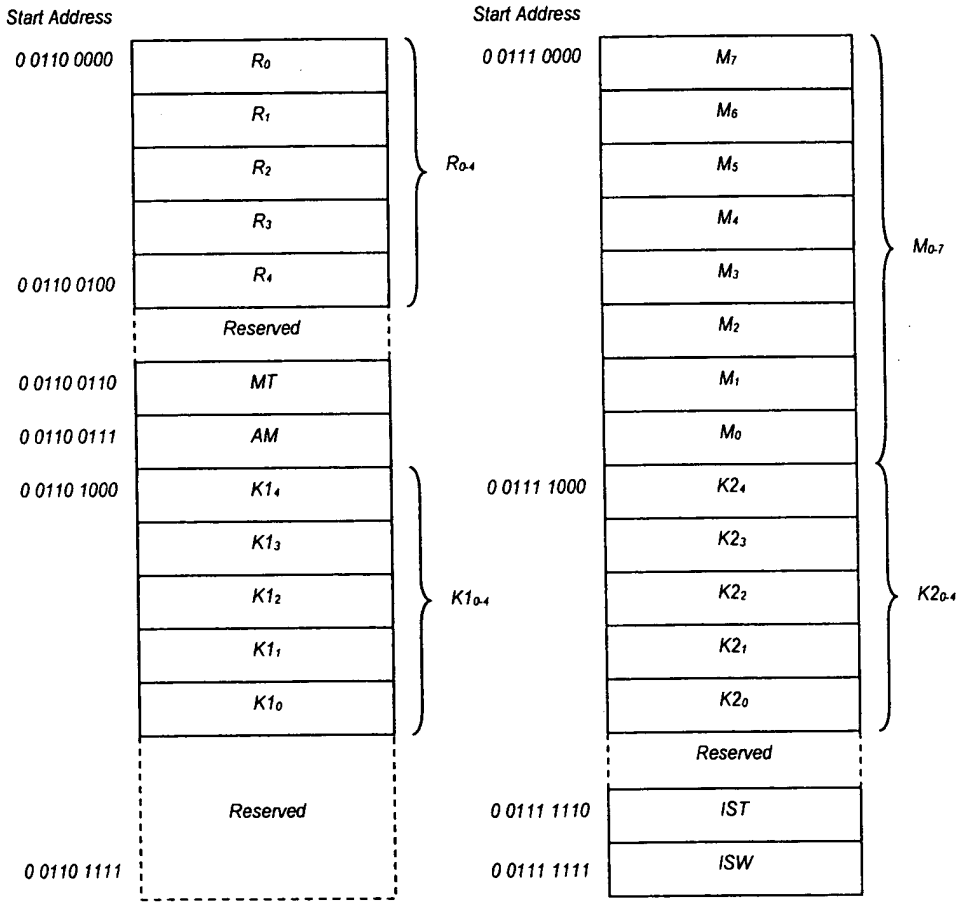

Start Address
0 0110 0000
R0
R1
R2
R3
R4
0 0110 0100
Reserved
0 0110 0110
MT
0 0110 0111
AM
0 0110 1000
K1 4
K1 3
K1 2
K1 1
K1 0
Reserved
0 0110 1111
R0-4
K1 0-4
Start Address
0 0111 0000
M7
M6
M5
M4
M3
M2
M1
M0
0 0111 1000
K2 4
K2 3
K2 2
K2 1
K2 0
Reserved
0 0111 1110
IST
0 0111 1111
ISW
M0-7
K2 0-4

FIG. 185

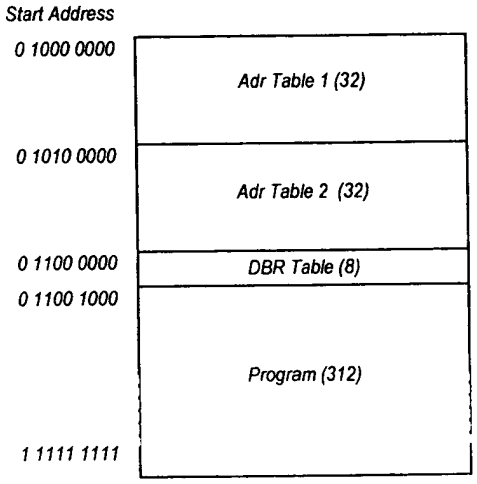

Start Address
0 1000 0000
Adr Table 1 (32)
0 1010 0000
Adr Table 2 (32)
0 1100 0000
DBR Table (8)
0 1100 1000
Program (312)
1 1111 1111

FIG. 186

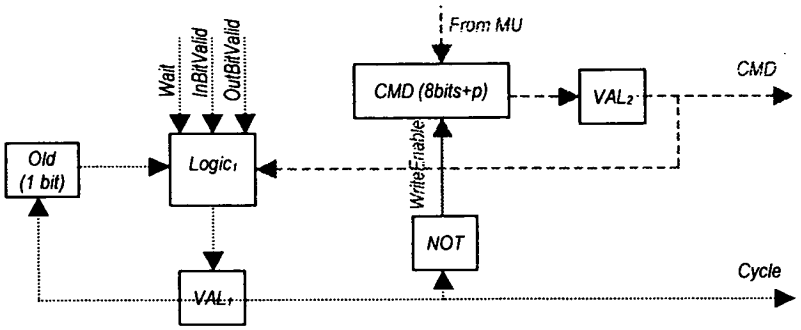


FIG. 187

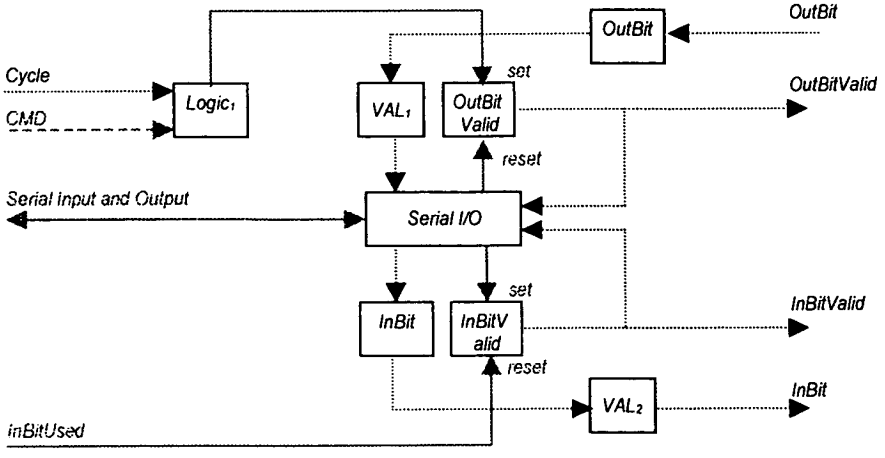


FIG. 188

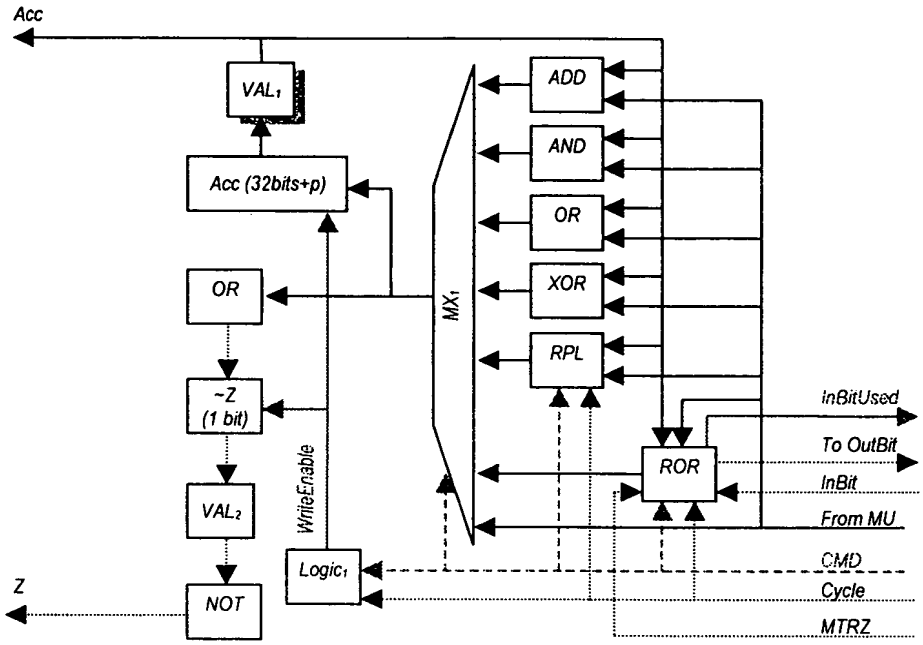

Acc
VAL1
Acc (32bits+p)
OR
~Z
(1 bit)
VAL2
NOT
Z
WriteEnable
Logic1
MX1
ADD
AND
OR
XOR
RPL
ROR
InBitUsed
To OutBit
InBit
From MU
CMD
Cycle
MTRZ

FIG. 189

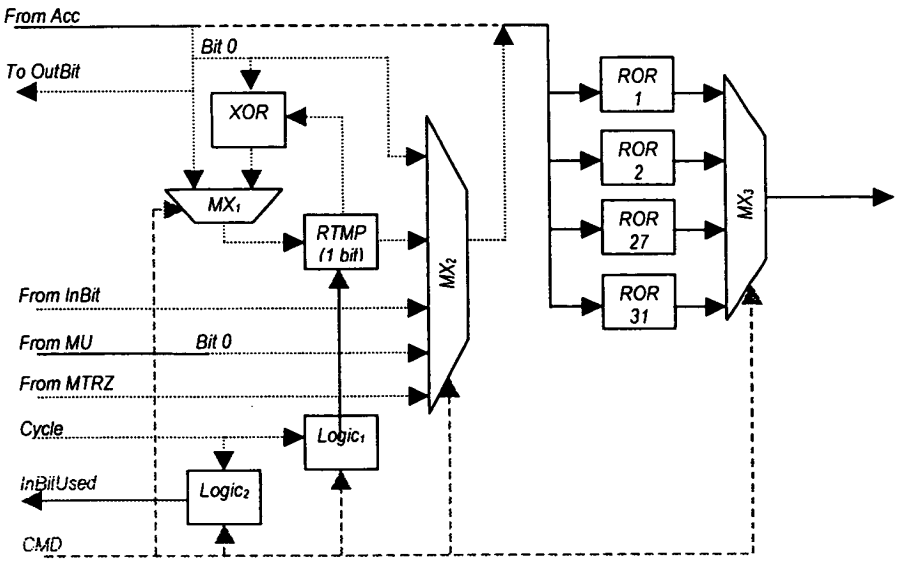

From Acc
Bit 0
To OutBit
XOR
MX1
RTMP
(1 bit)
MX2
ROR
1
ROR
2
ROR
27
ROR
31
MX3
From InBit
From MU
Bit 0
From MTRZ
Cycle
Logic1
InBitUsed
Logic2
CMD

FIG. 190

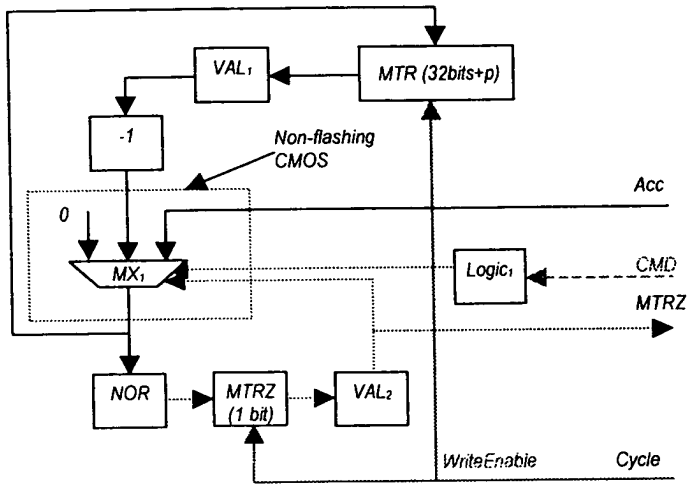

VAL1
MTR (32bits+p)
-1
Non-flashing
CMOS
Acc
0
MX1
Logic1
CMD
MTRZ
NOR
MTRZ
(1 bit)
VAL2
WriteEnable
Cycle

FIG. 191

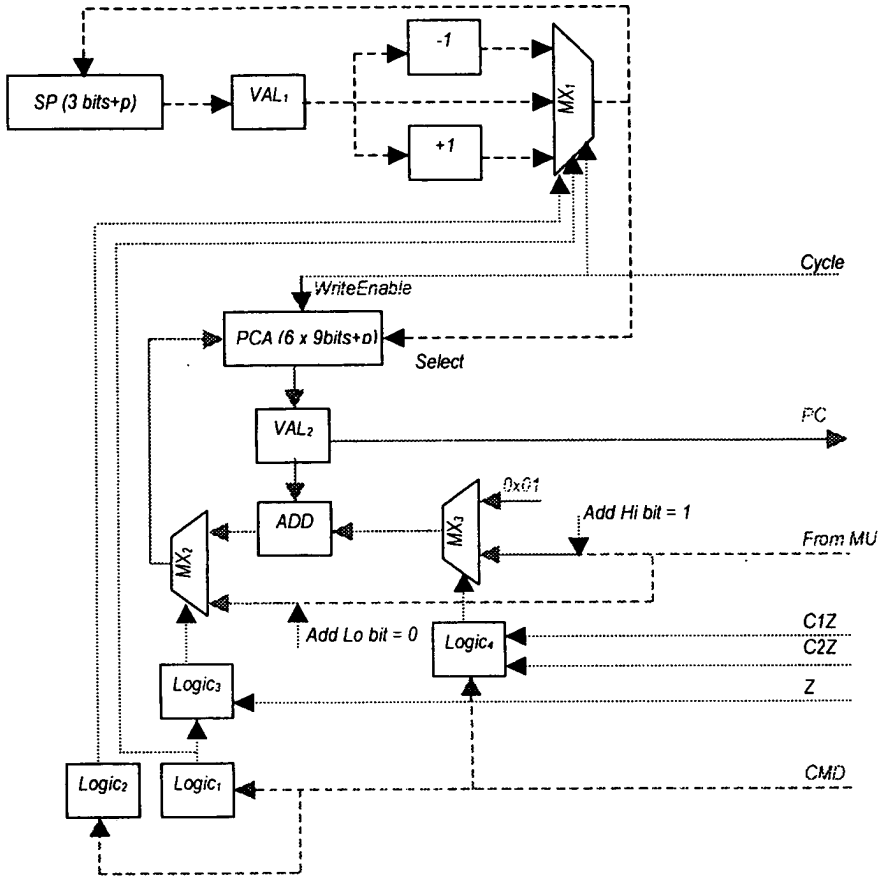

SP (3 bits+p)
VAL1
-1
+1
MX1
Cycle
WriteEnable
PCA (6 x 9bits+p)
Select
VAL2
PC
0x01
ADD
MX3
Add Hi bit = 1
From MU
MX2
Add Lo bit = 0
Logic4
C1Z
C2Z
Z
Logic3
CMD
Logic2
Logic1

FIG. 192

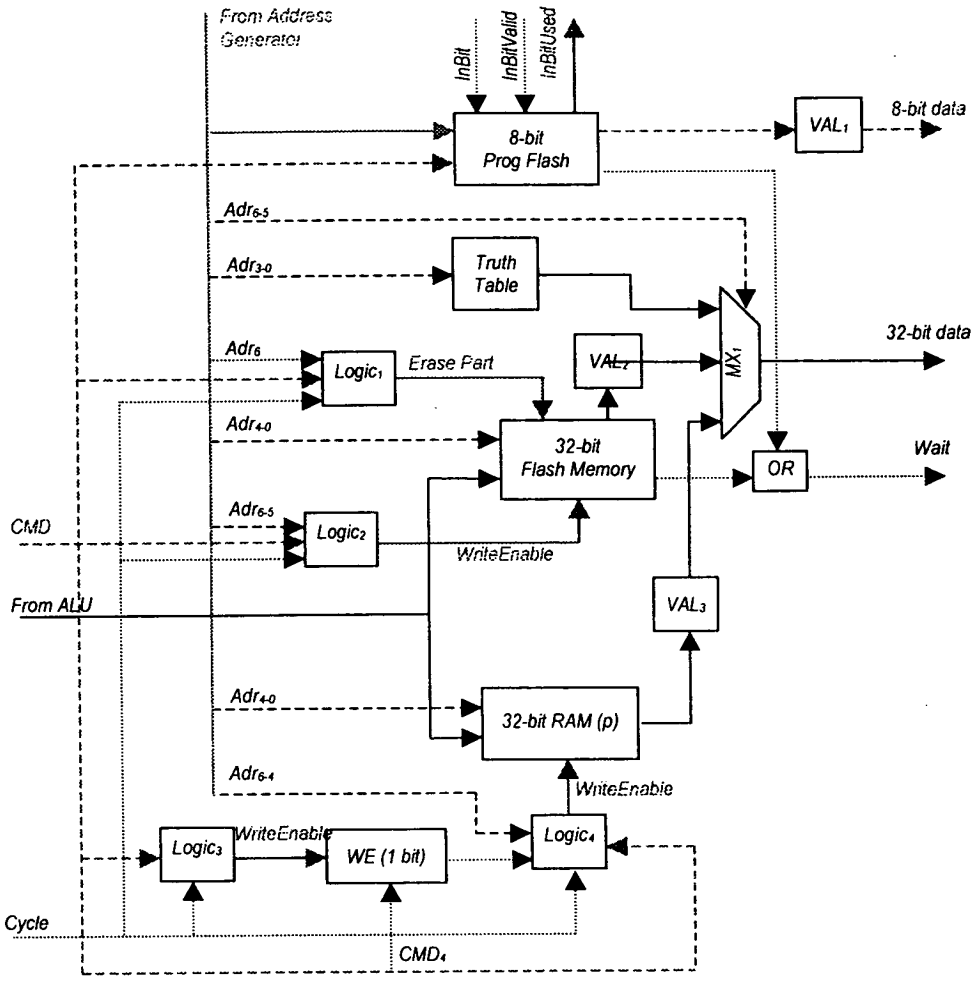

From Address Generator
InBit
InBitValid
InBitUsed
8-bit Prog Flash
VAL1
8-bit data
Adr6-5
Adr3-0
Truth Table
Adr6
Logic1
Erase Part
VAL2
MX1
32-bit data
Adr4-0
32-bit Flash Memory
OR
Wait
CMD
Adr6-5
Logic2
WriteEnable
From ALU
VAL3
Adr4-0
32-bit RAM (p)
Adr6-4
WriteEnable
Logic3
WriteEnable
WE (1 bit)
Logic4
Cycle
CMD4

FIG. 193

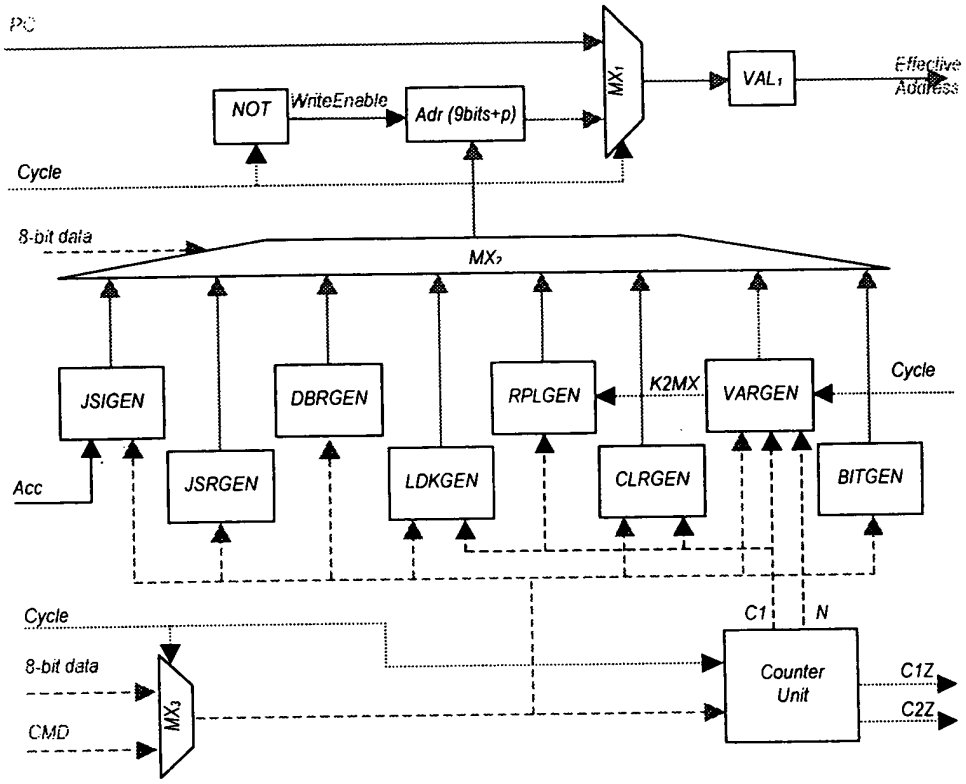

PC
NOT
WriteEnable
Adr (9bits+p)
MX1
VAL1
Effective Address
Cycle
8-bit data
MX2
JSIGEN
DBRGEN
RPLGEN
K2MX
VARGEN
Cycle
Acc
JSRGEN
LDKGEN
CLRGEN
BITGEN
Cycle
C1
N
8-bit data
CMD
MX3
Counter Unit
C1Z
C2Z

FIG. 194

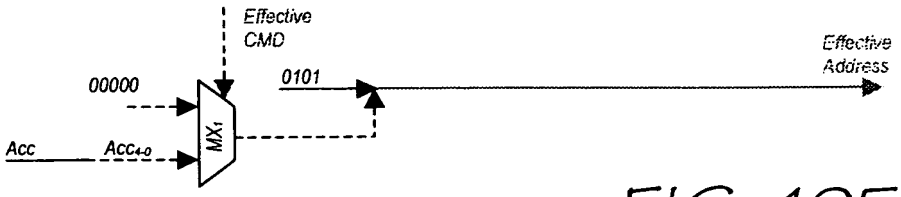


FIG. 195

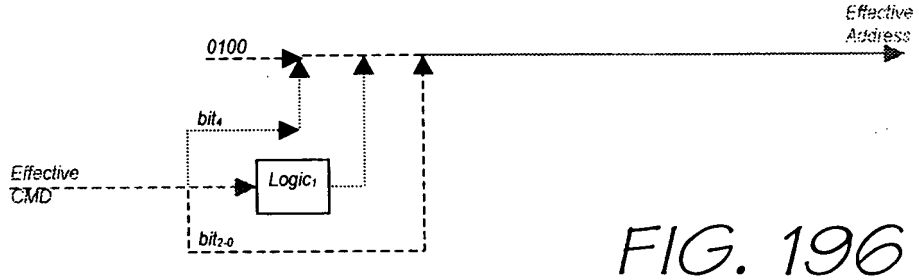


FIG. 196

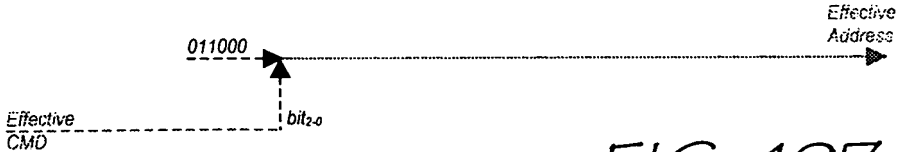


FIG. 197

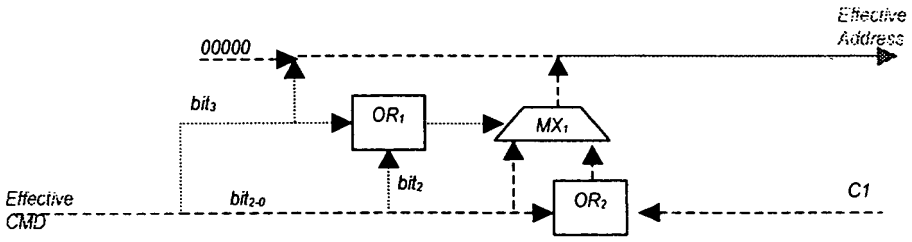


FIG. 198

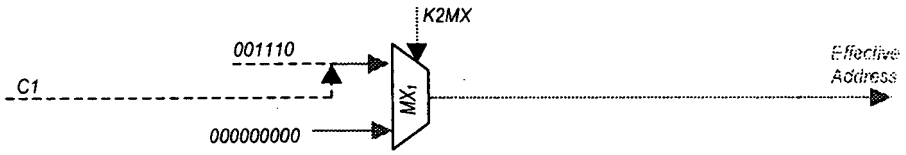


FIG. 199

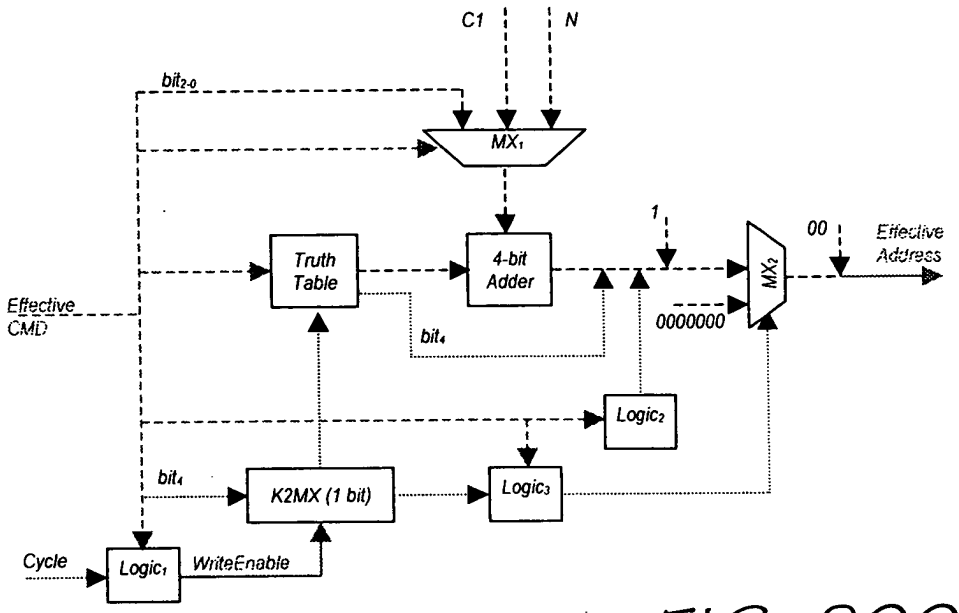


FIG. 200

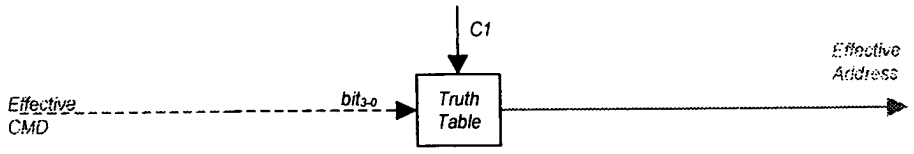


FIG. 201

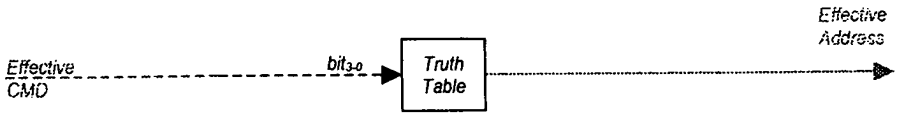


FIG. 202

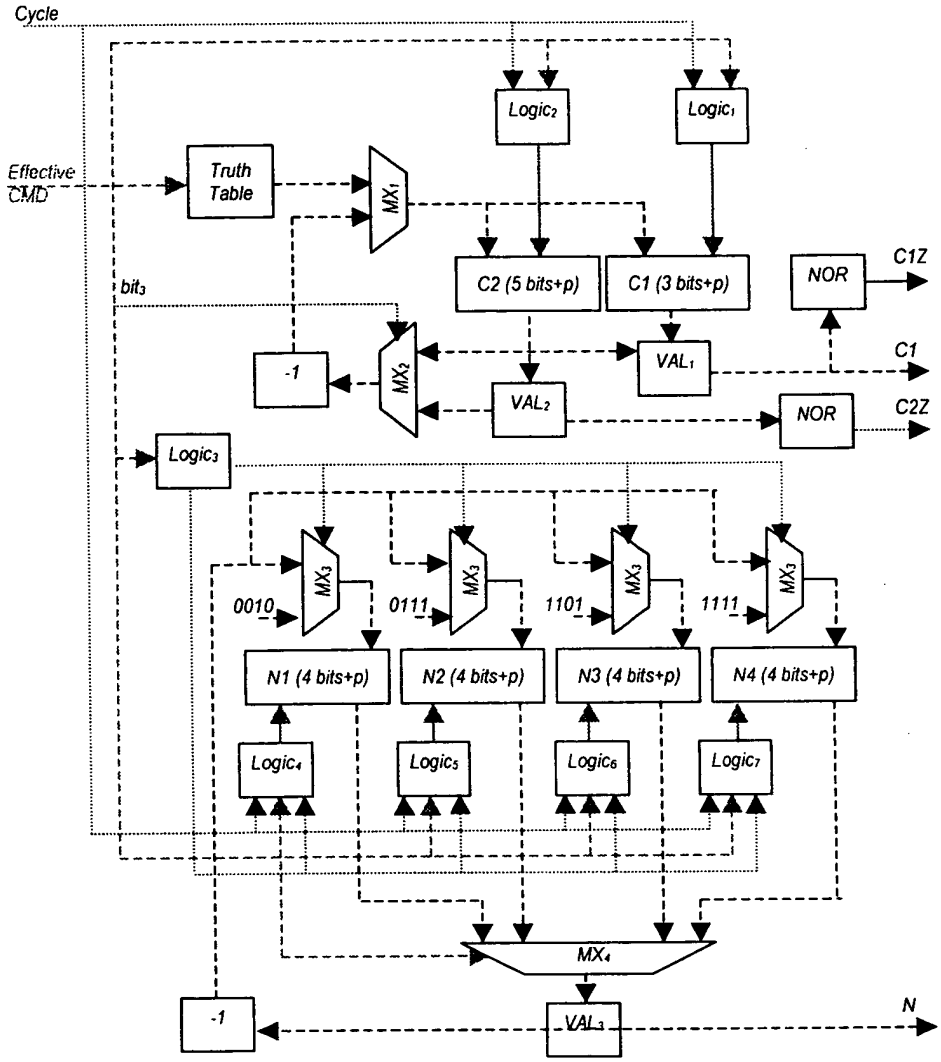

Cycle
Effective CMD
Truth Table
MX1
Logic2
Logic1
C2 (5 bits+p)
C1 (3 bits+p)
NOR
C1Z
bit3
-1
MX2
VAL1
C1
VAL2
NOR
C2Z
Logic3
MX3
0010
0111
1101
1111
N1 (4 bits+p)
N2 (4 bits+p)
N3 (4 bits+p)
N4 (4 bits+p)
Logic4
Logic5
Logic6
Logic7
MX4
-1
VAL3
N

FIG. 203

705

Data Type	Bits
Factory code	16
Batch number	32
Serial number	48
Manufacturing date	16
Media length	24
Media type	8
Preprinted media length	16
Cyan ink viscosity	8
Magenta ink viscosity	8
Yellow ink viscosity	8
Cyan drop volume	8
Magenta drop volume	8
Yellow drop volume	8
Cyan ink color	24
Magenta ink color	24
Yellow ink color	24
Remaining-media length indicator	16
Authentication key	128
Copyrightable bit pattern	512
Reserved for camera use	88
Total	**1024**

728

FIG. 204

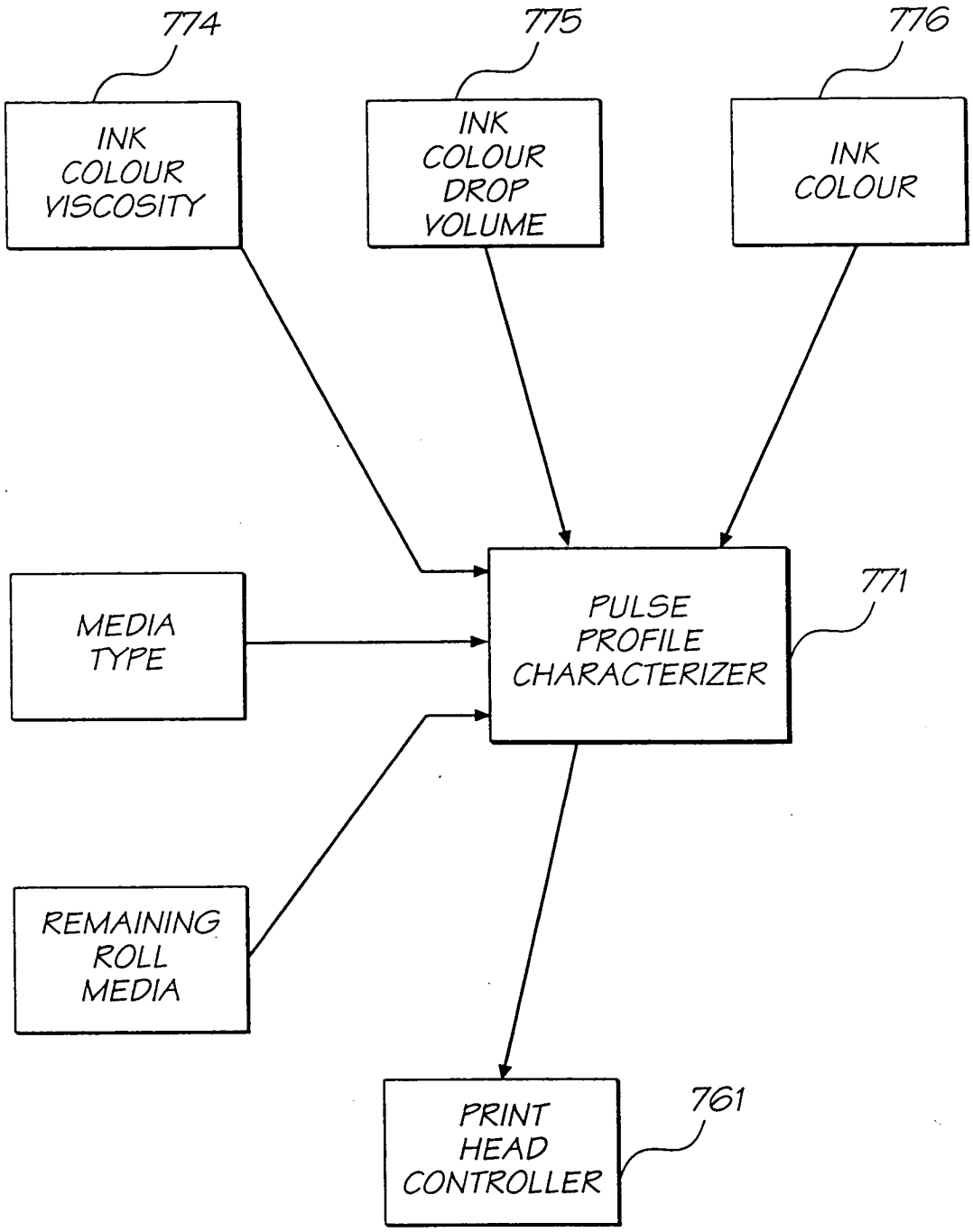

774
775
776
INK COLOUR VISCOSITY
INK COLOUR DROP VOLUME
INK COLOUR
MEDIA TYPE
PULSE PROFILE CHARACTERIZER
771
REMAINING ROLL MEDIA
PRINT HEAD CONTROLLER
761

FIG. 205

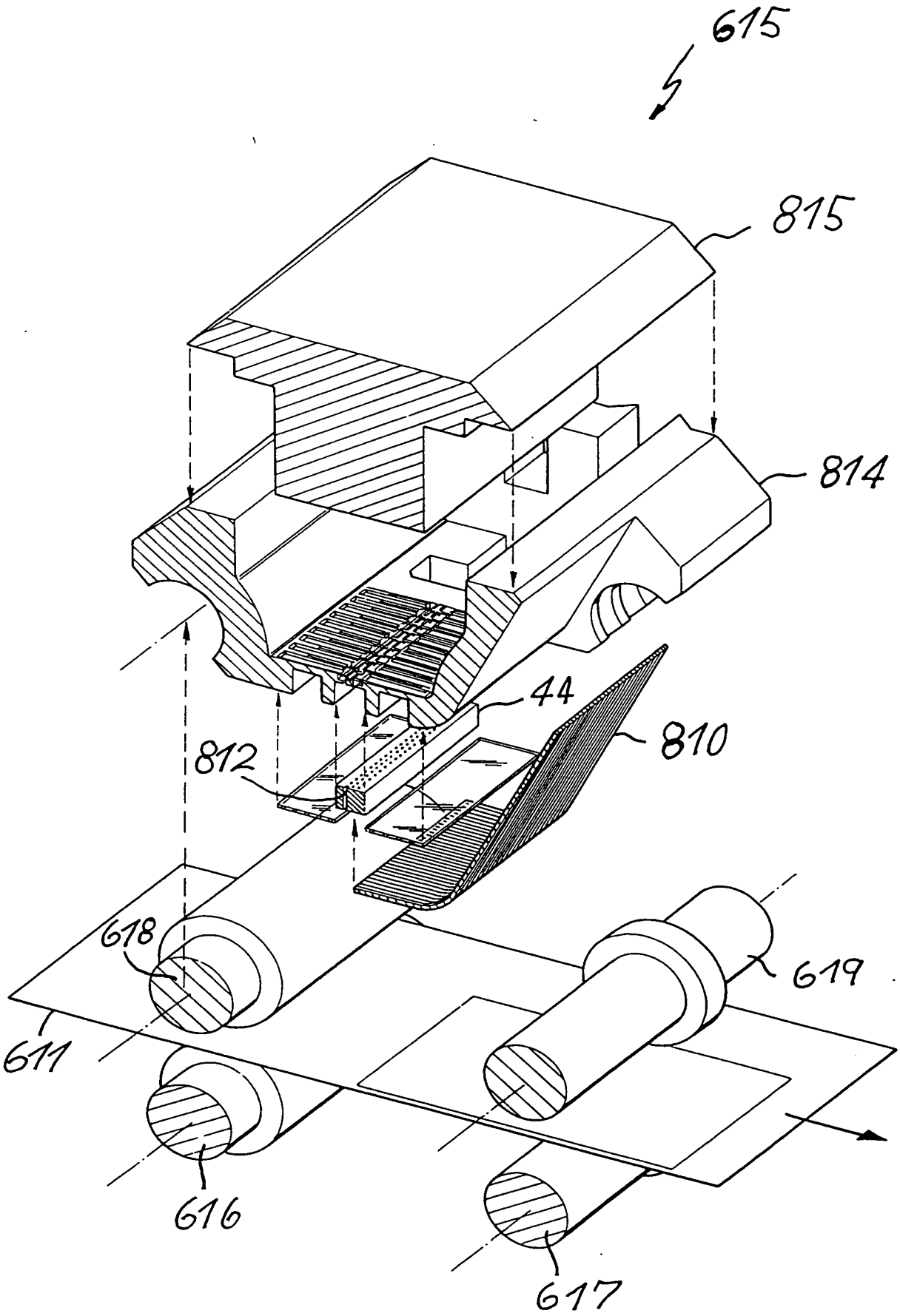


FIG. 206

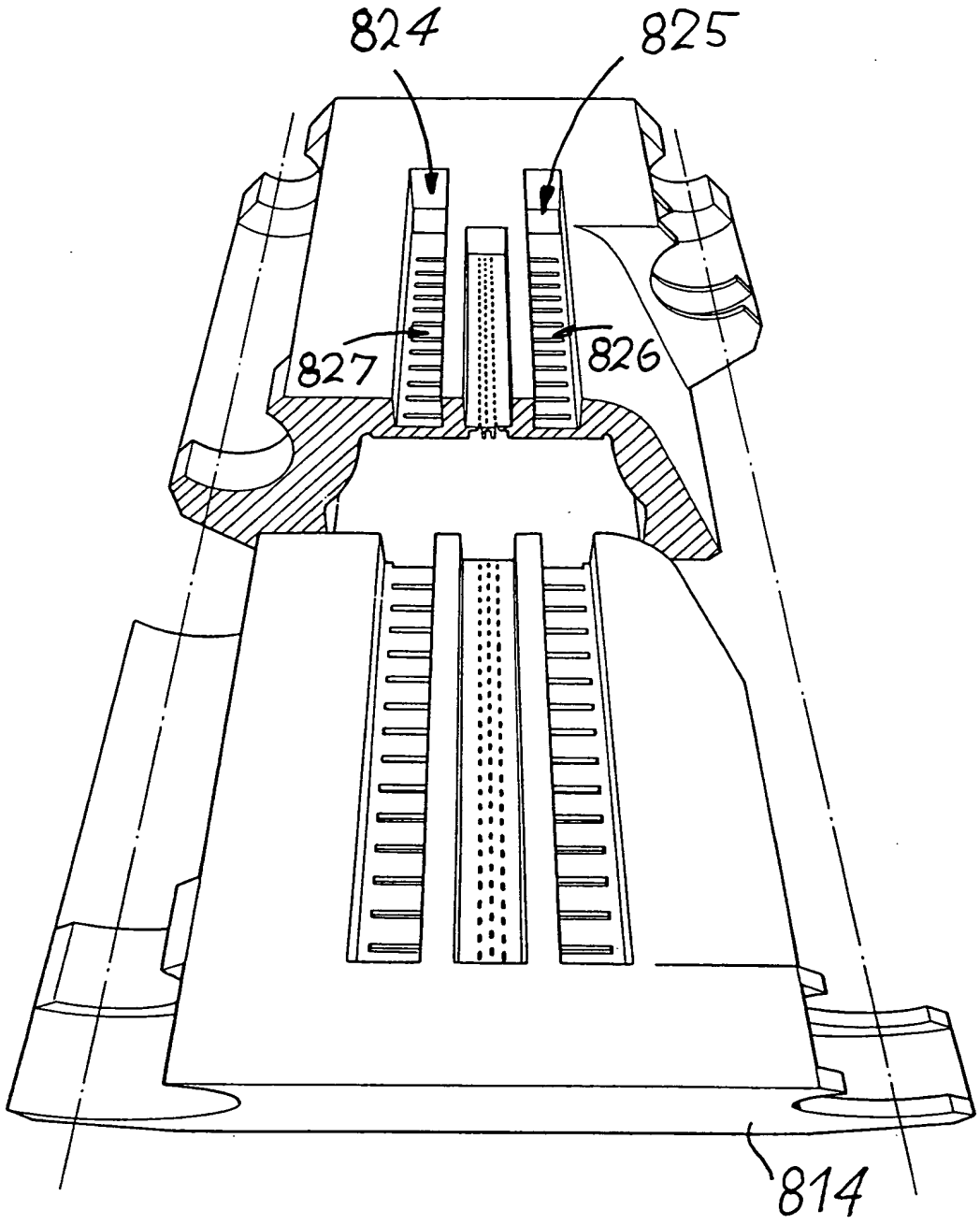


FIG. 207

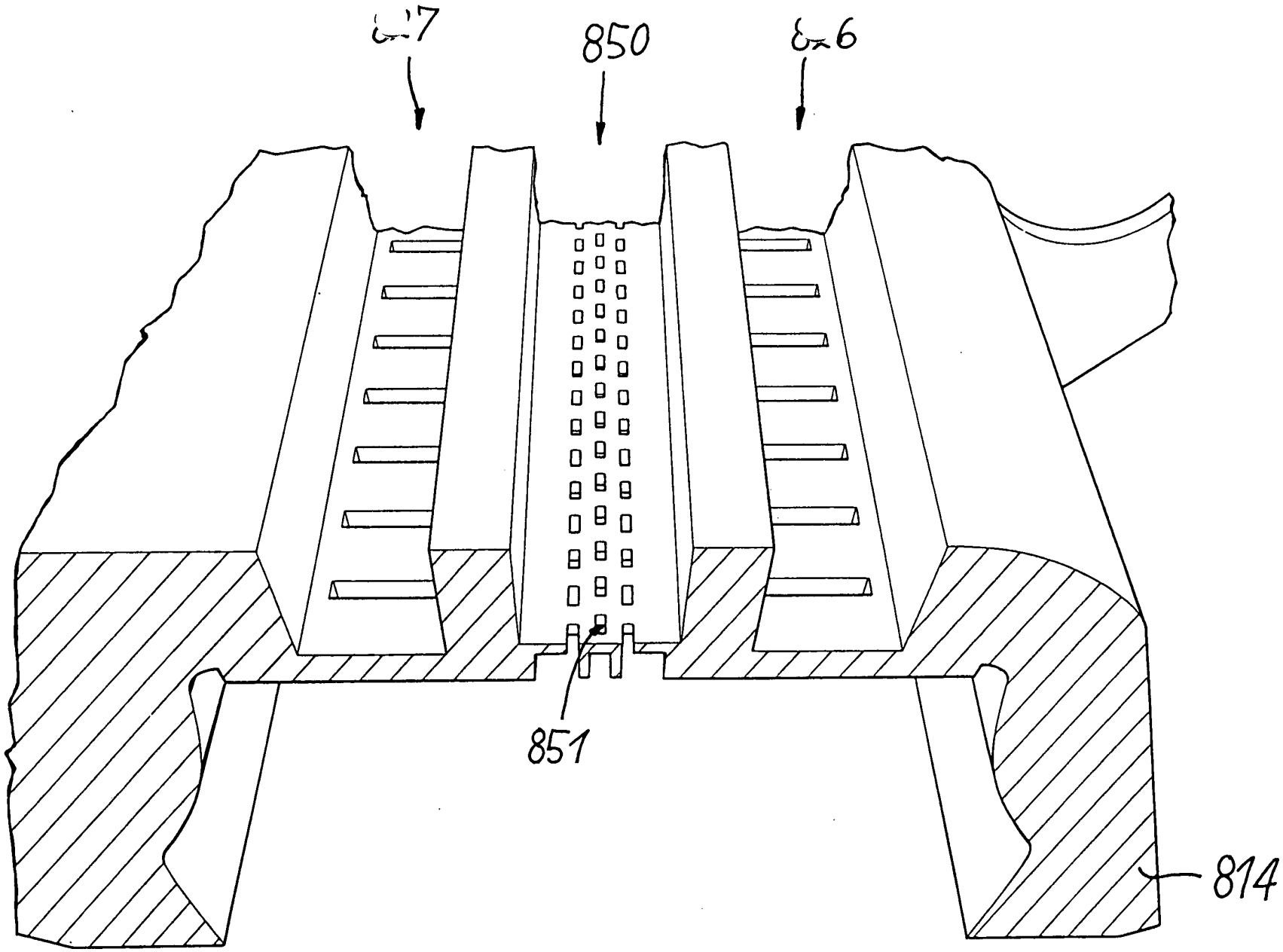


FIG. 208

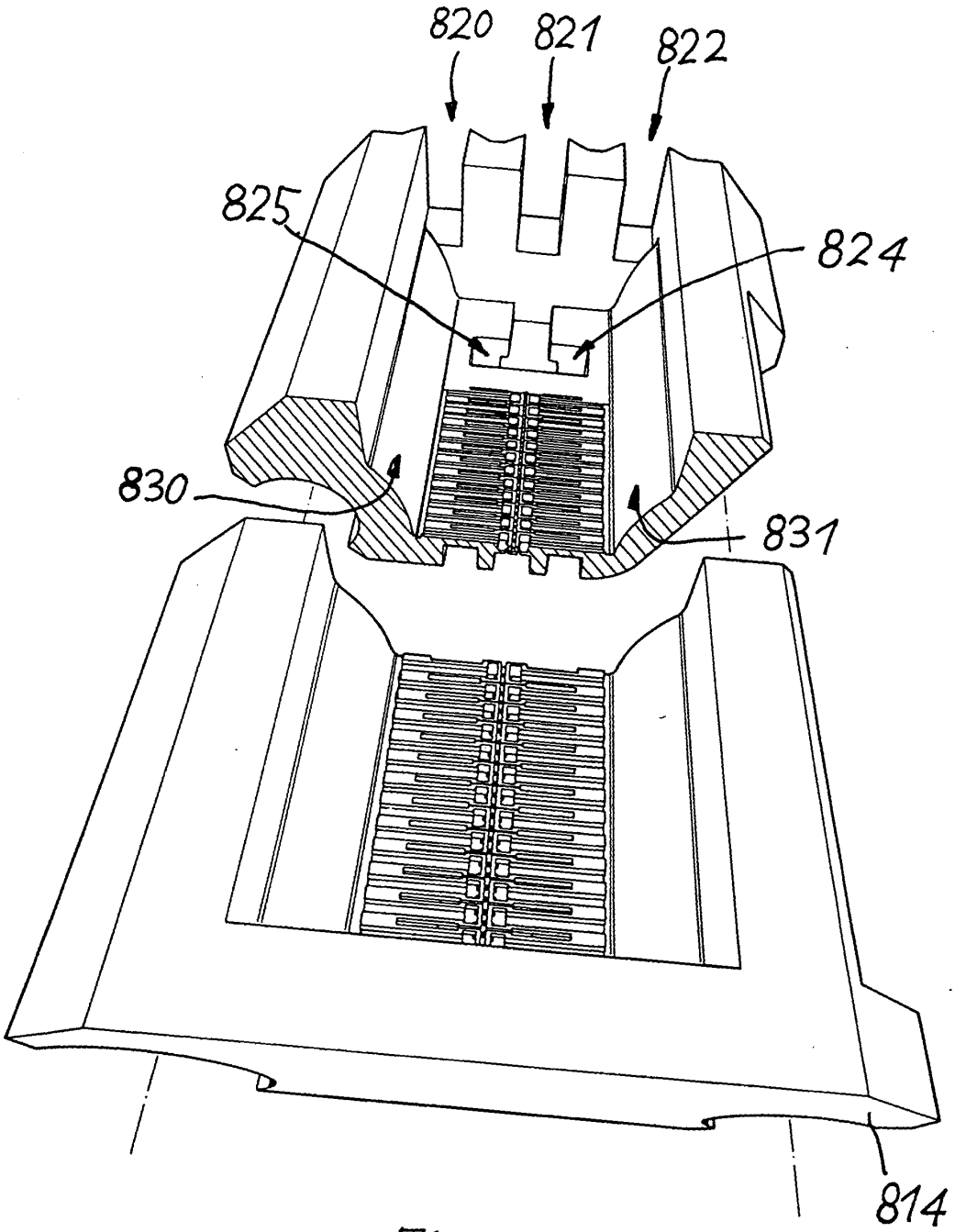


FIG. 209

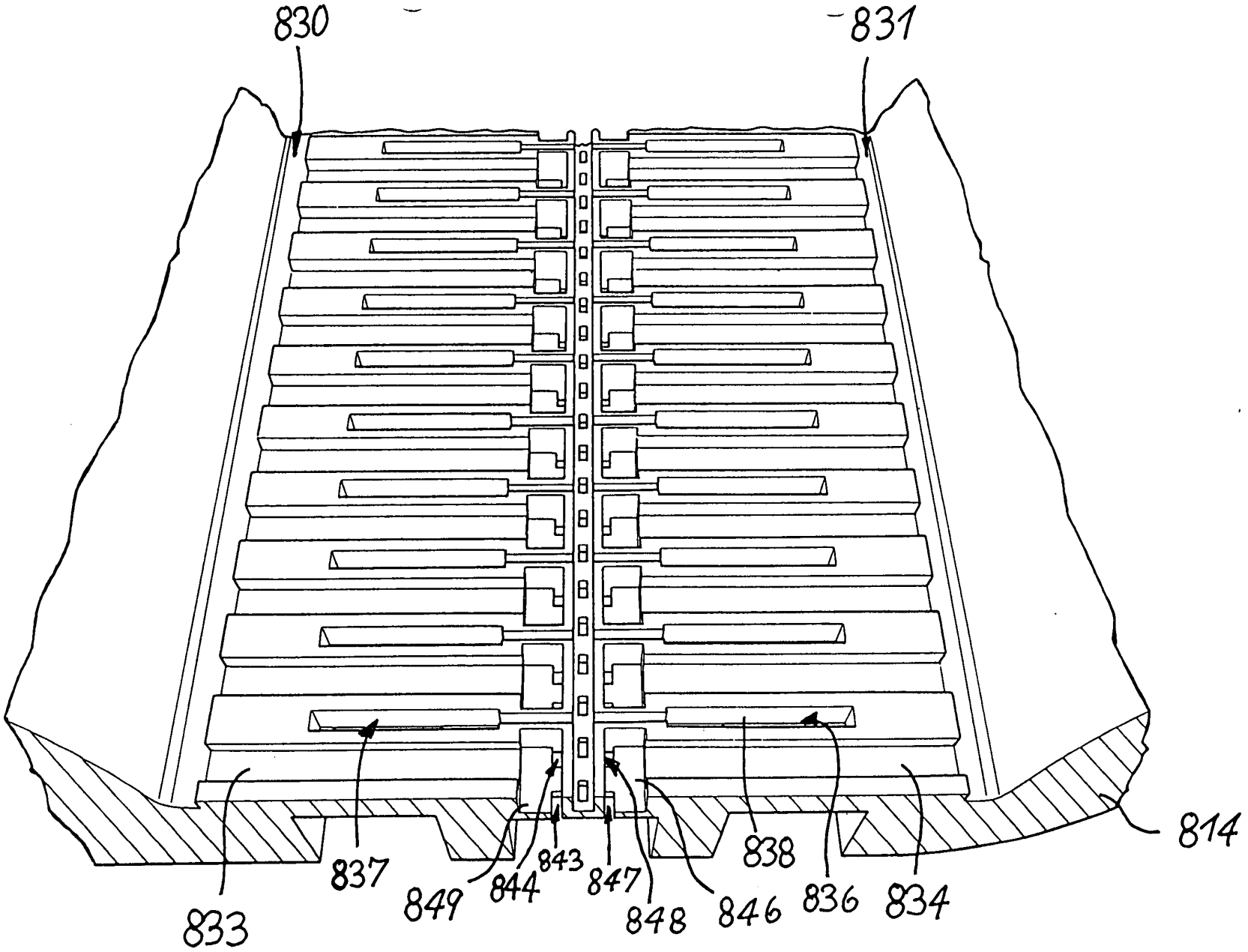


FIG. 210

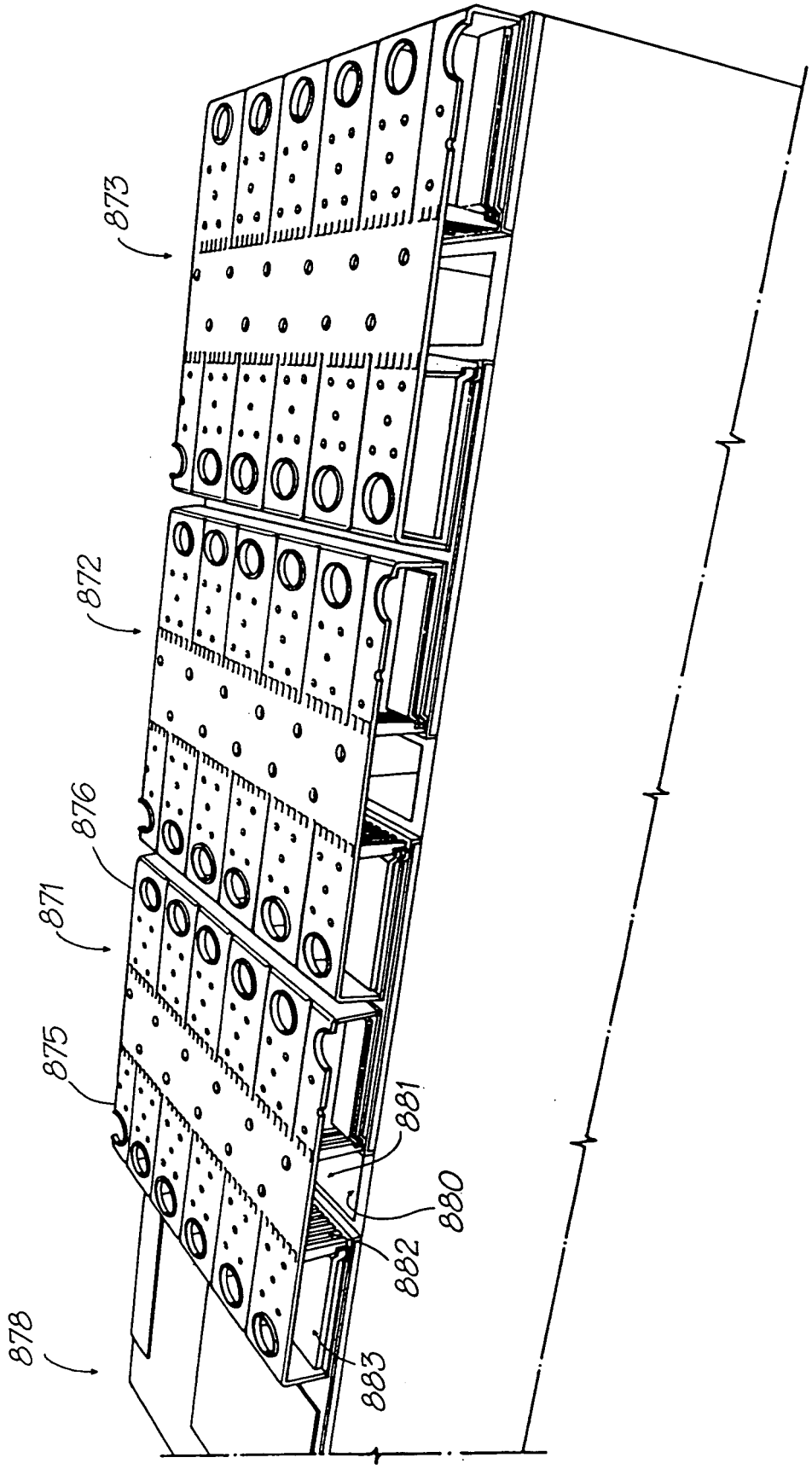


FIG. 211

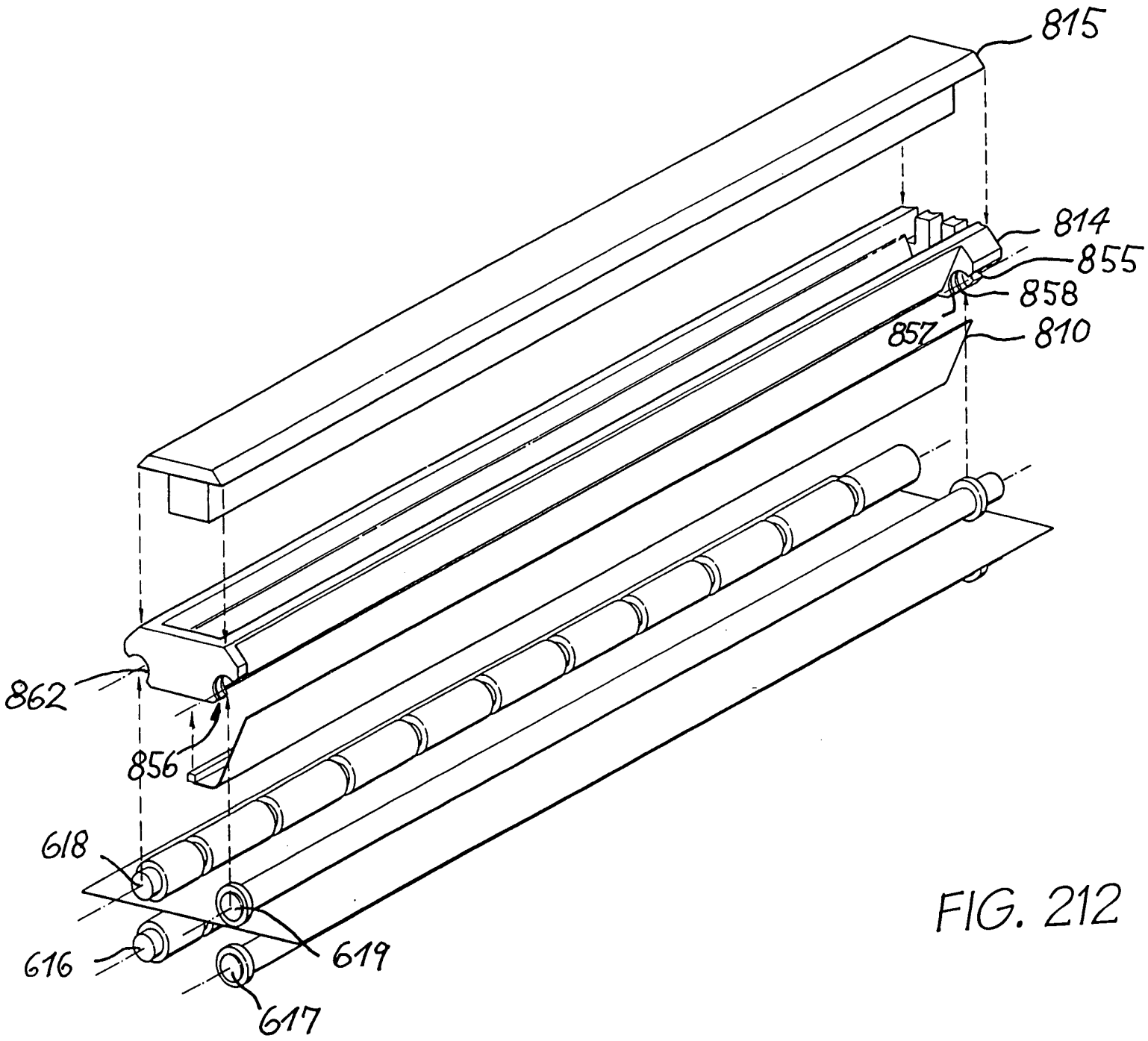


FIG. 212

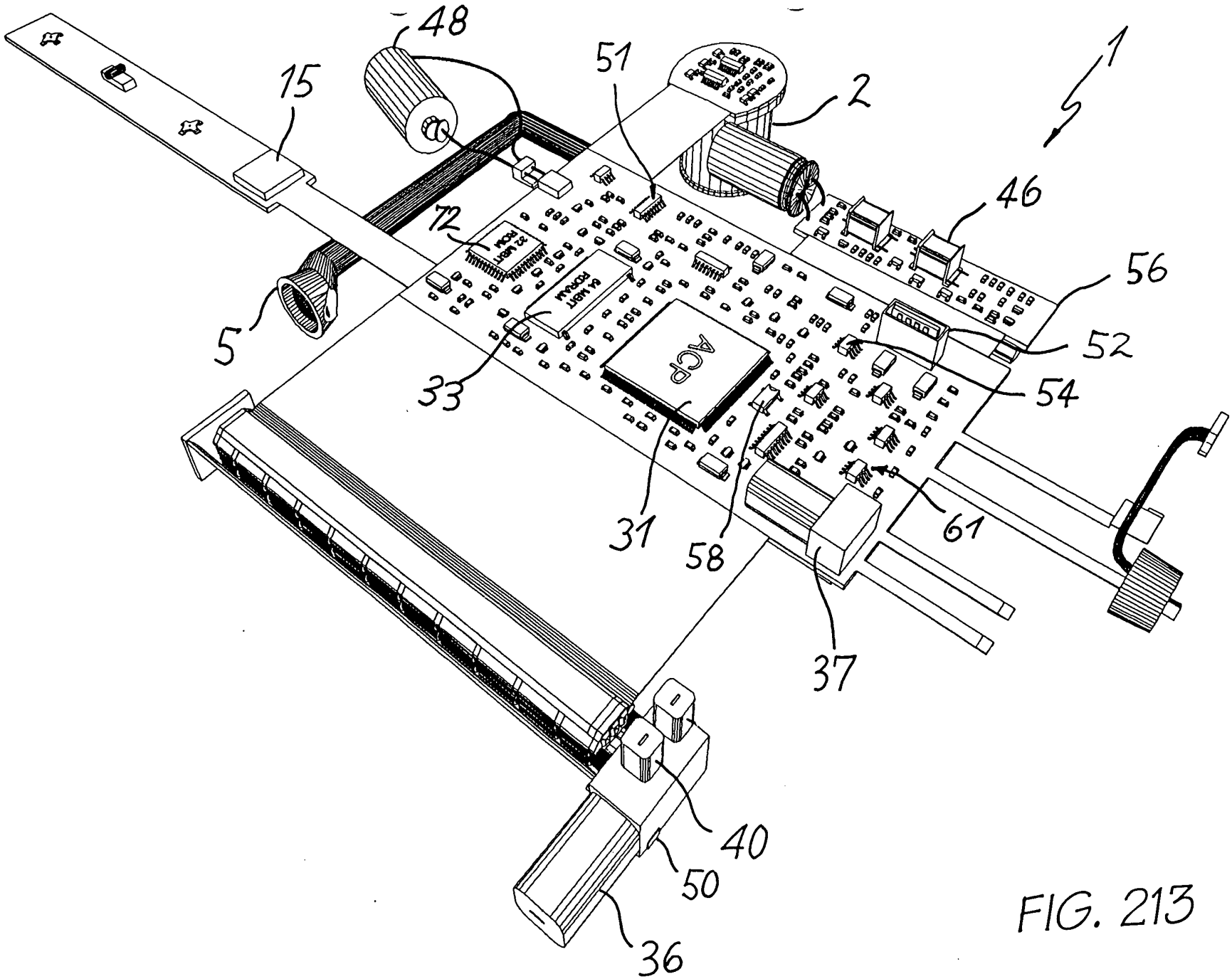


FIG. 213

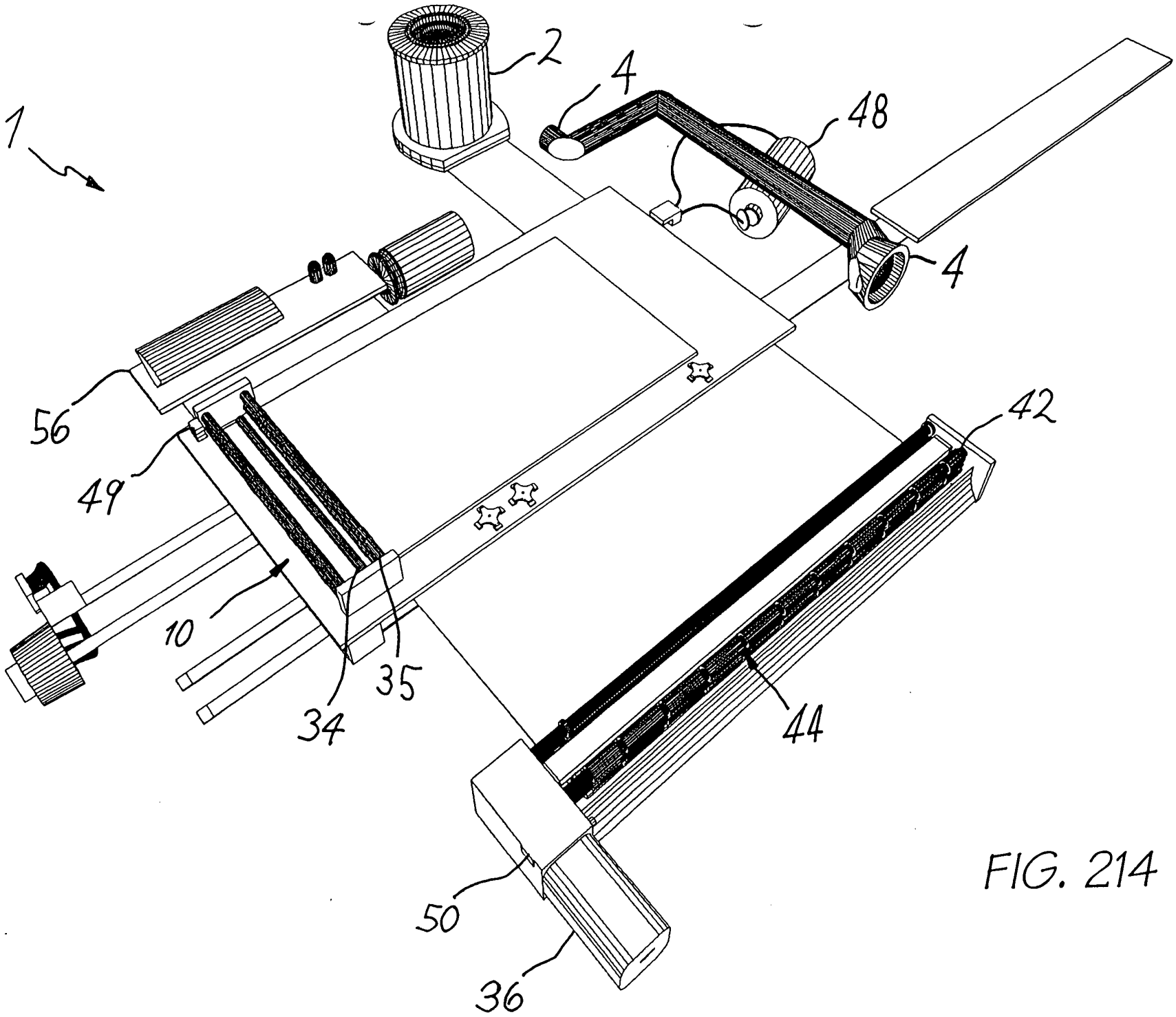


FIG. 214

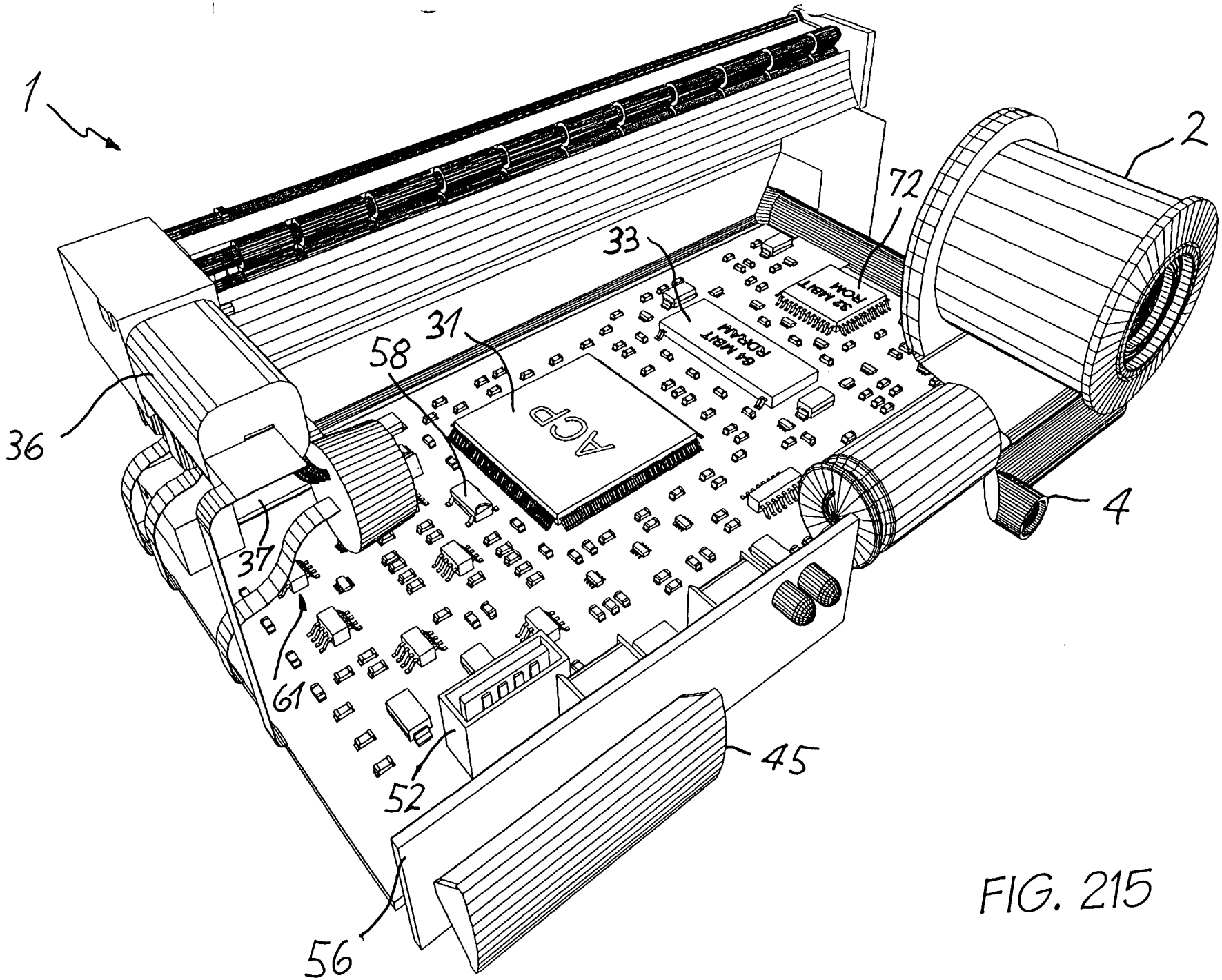


FIG. 215

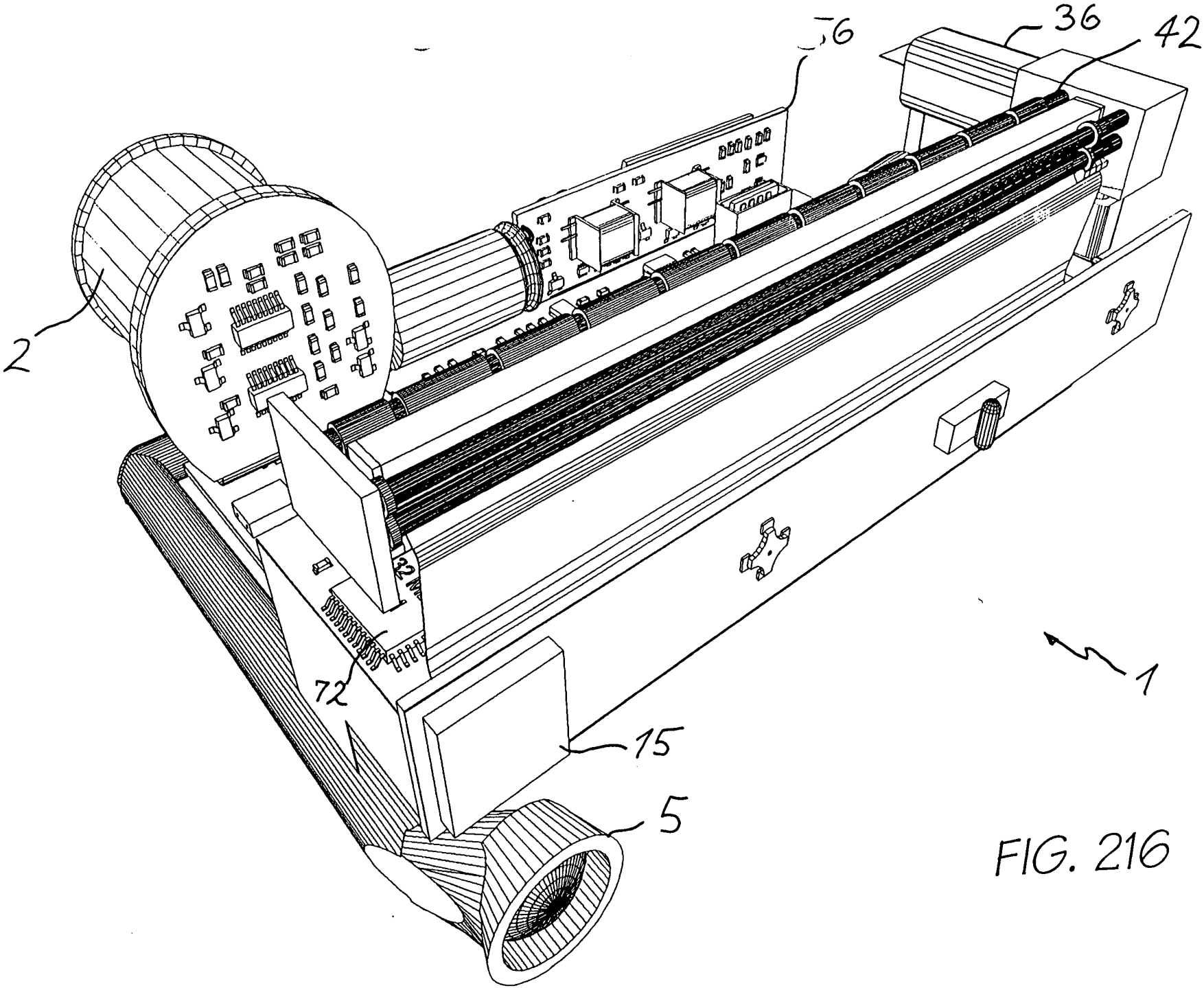


FIG. 216

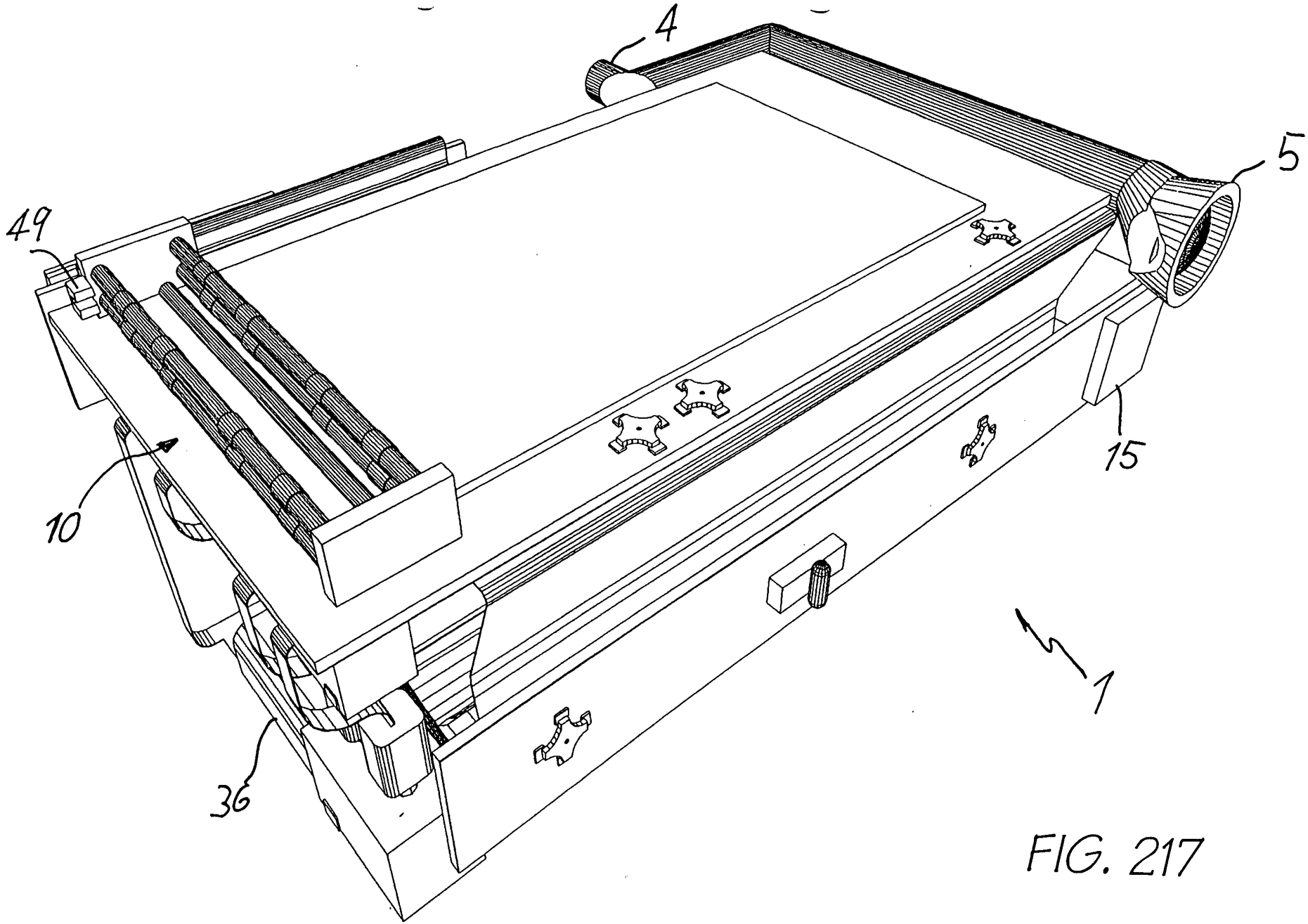


FIG. 217

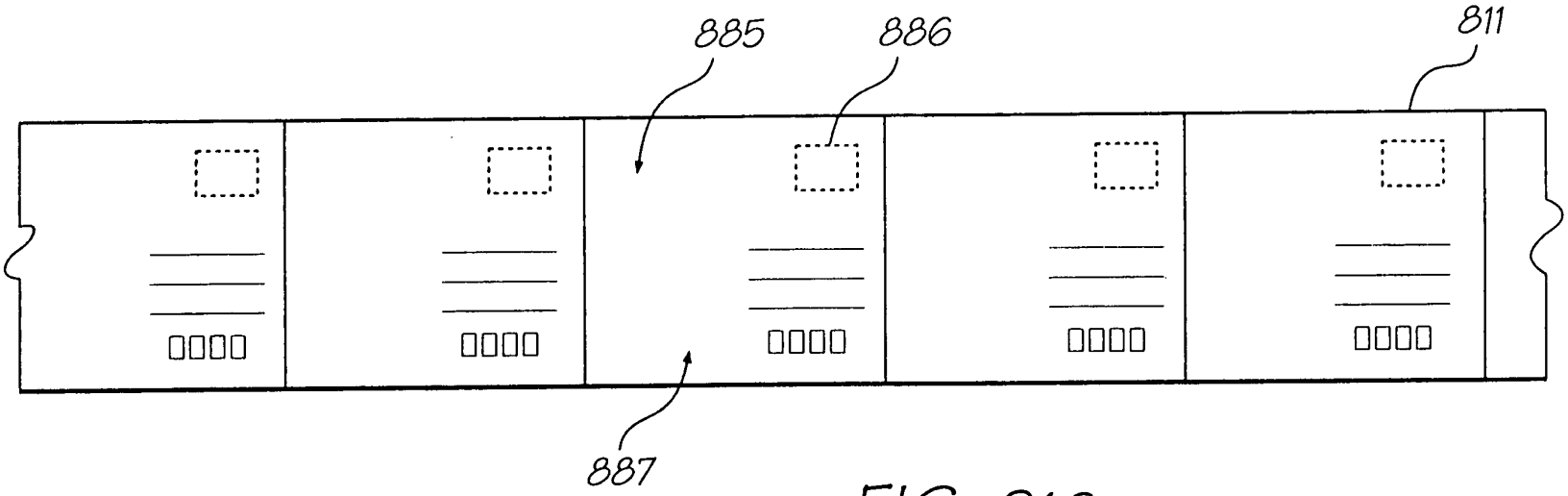


FIG. 218

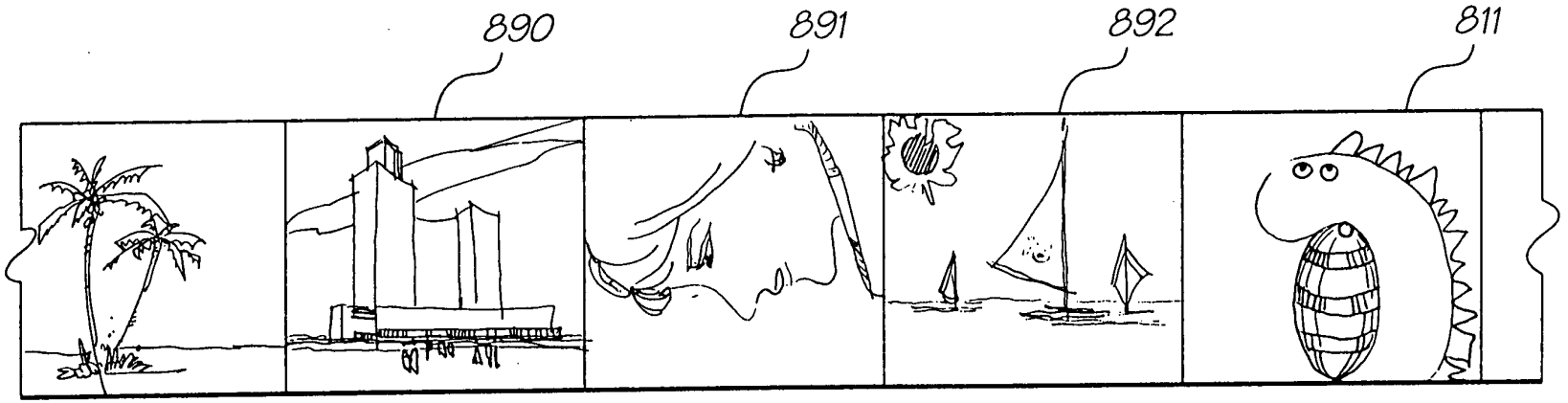


FIG. 219

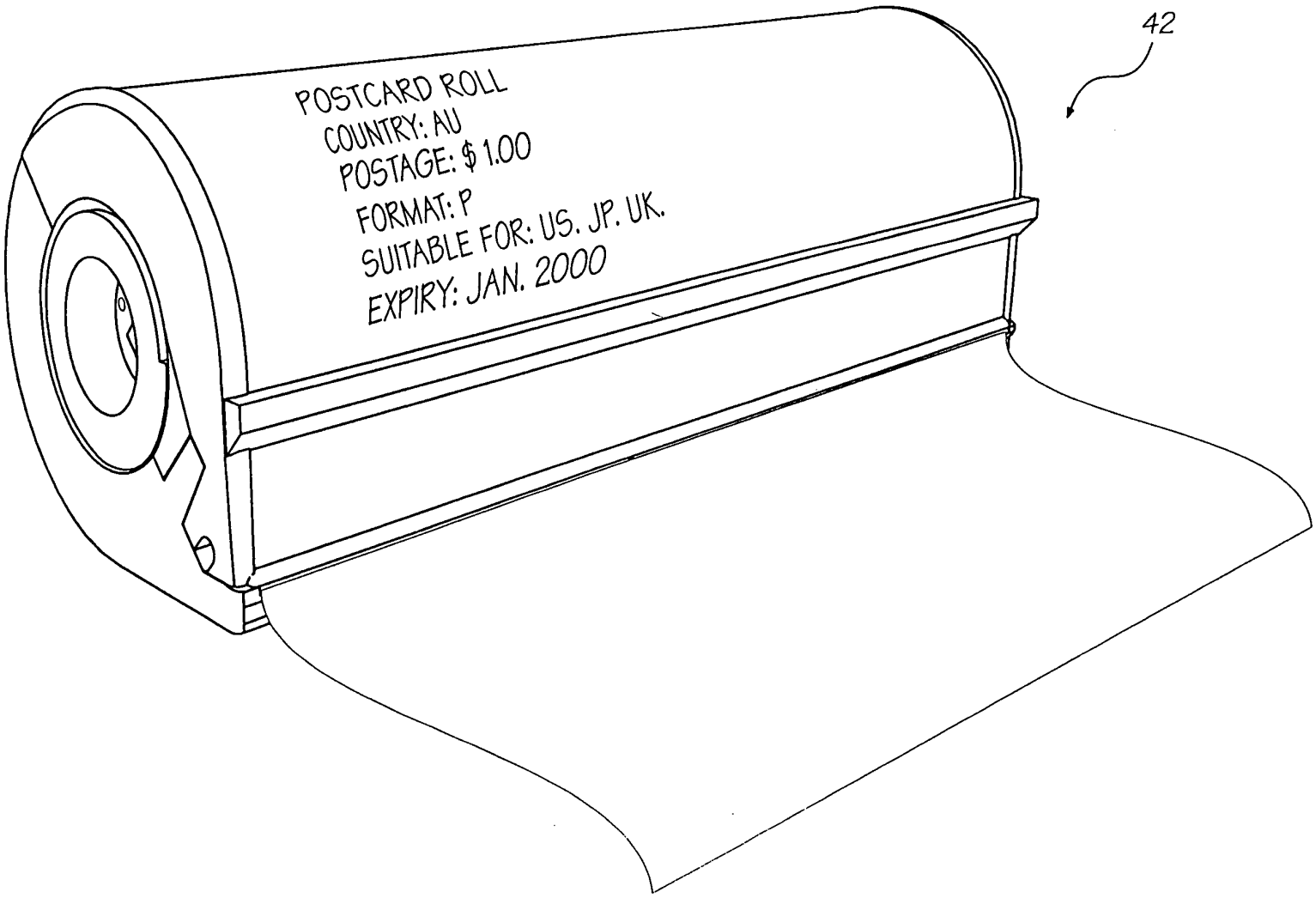

42
POSTCARD ROLL
COUNTRY: AU
POSTAGE: $1.00
FORMAT: P
SUITABLE FOR: US. JP. UK.
EXPIRY: JAN. 2000

FIG. 220

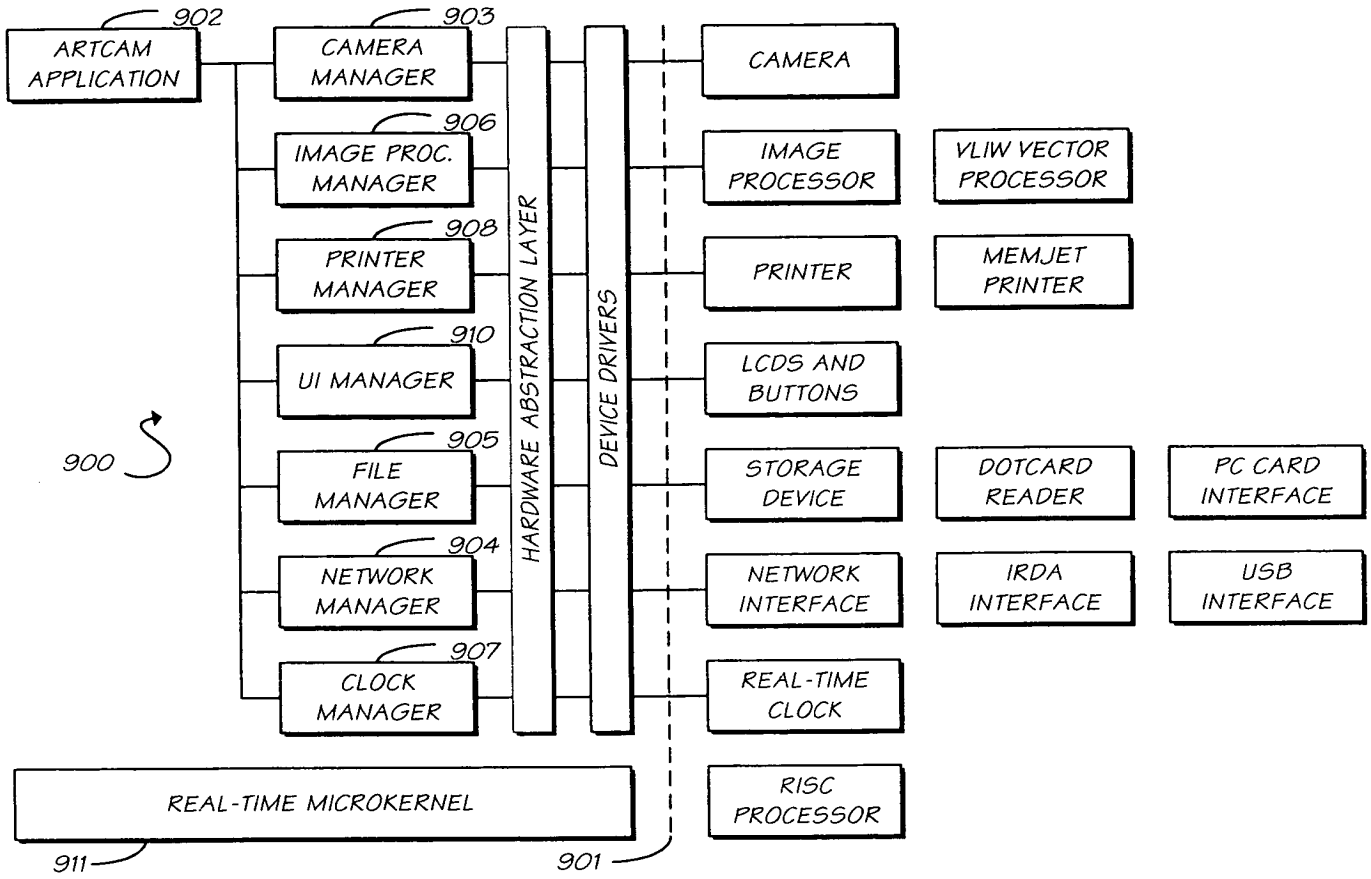

902
ARTCAM APPLICATION
903
CAMERA MANAGER
906
IMAGE PROC. MANAGER
908
PRINTER MANAGER
910
UI MANAGER
905
FILE MANAGER
904
NETWORK MANAGER
907
CLOCK MANAGER
900
HARDWARE ABSTRACTION LAYER
DEVICE DRIVERS
CAMERA
IMAGE PROCESSOR
VLIW VECTOR PROCESSOR
PRINTER
MEMJET PRINTER
LCDS AND BUTTONS
STORAGE DEVICE
DOTCARD READER
PC CARD INTERFACE
NETWORK INTERFACE
IRDA INTERFACE
USB INTERFACE
REAL-TIME CLOCK
RISC PROCESSOR
REAL-TIME MICROKERNEL
911
901

FIG. 221

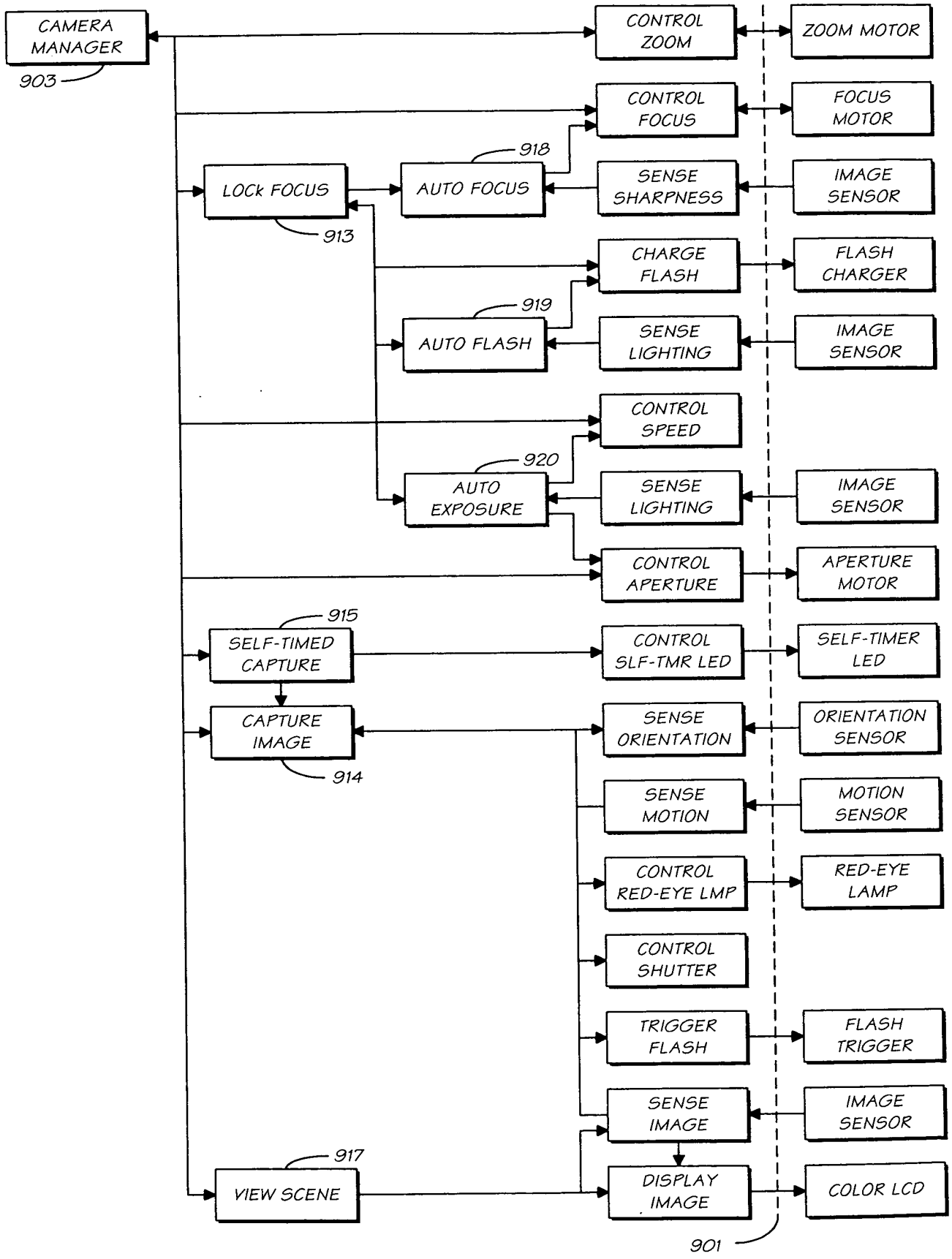

CAMERA MANAGER
903
CONTROL ZOOM
ZOOM MOTOR
CONTROL FOCUS
FOCUS MOTOR
LOCK FOCUS
913
AUTO FOCUS
918
SENSE SHARPNESS
IMAGE SENSOR
CHARGE FLASH
FLASH CHARGER
AUTO FLASH
919
SENSE LIGHTING
IMAGE SENSOR
CONTROL SPEED
AUTO EXPOSURE
920
SENSE LIGHTING
IMAGE SENSOR
CONTROL APERTURE
APERTURE MOTOR
SELF-TIMED CAPTURE
915
CONTROL SLF-TMR LED
SELF-TIMER LED
CAPTURE IMAGE
914
SENSE ORIENTATION
ORIENTATION SENSOR
SENSE MOTION
MOTION SENSOR
CONTROL RED-EYE LMP
RED-EYE LAMP
CONTROL SHUTTER
TRIGGER FLASH
FLASH TRIGGER
SENSE IMAGE
IMAGE SENSOR
VIEW SCENE
917
DISPLAY IMAGE
COLOR LCD
901

FIG. 222

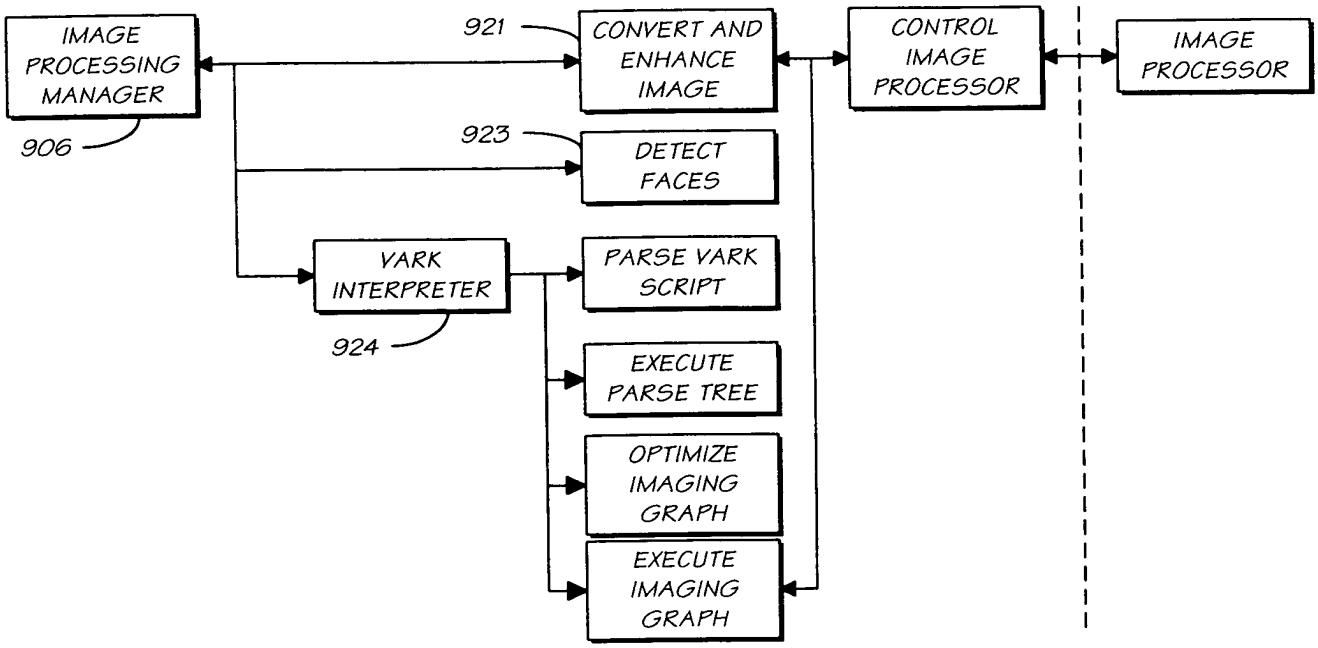

IMAGE PROCESSING MANAGER
906
921
CONVERT AND ENHANCE IMAGE
923
DETECT FACES
VARK INTERPRETER
924
PARSE VARK SCRIPT
EXECUTE PARSE TREE
OPTIMIZE IMAGING GRAPH
EXECUTE IMAGING GRAPH
CONTROL IMAGE PROCESSOR
IMAGE PROCESSOR

FIG. 223

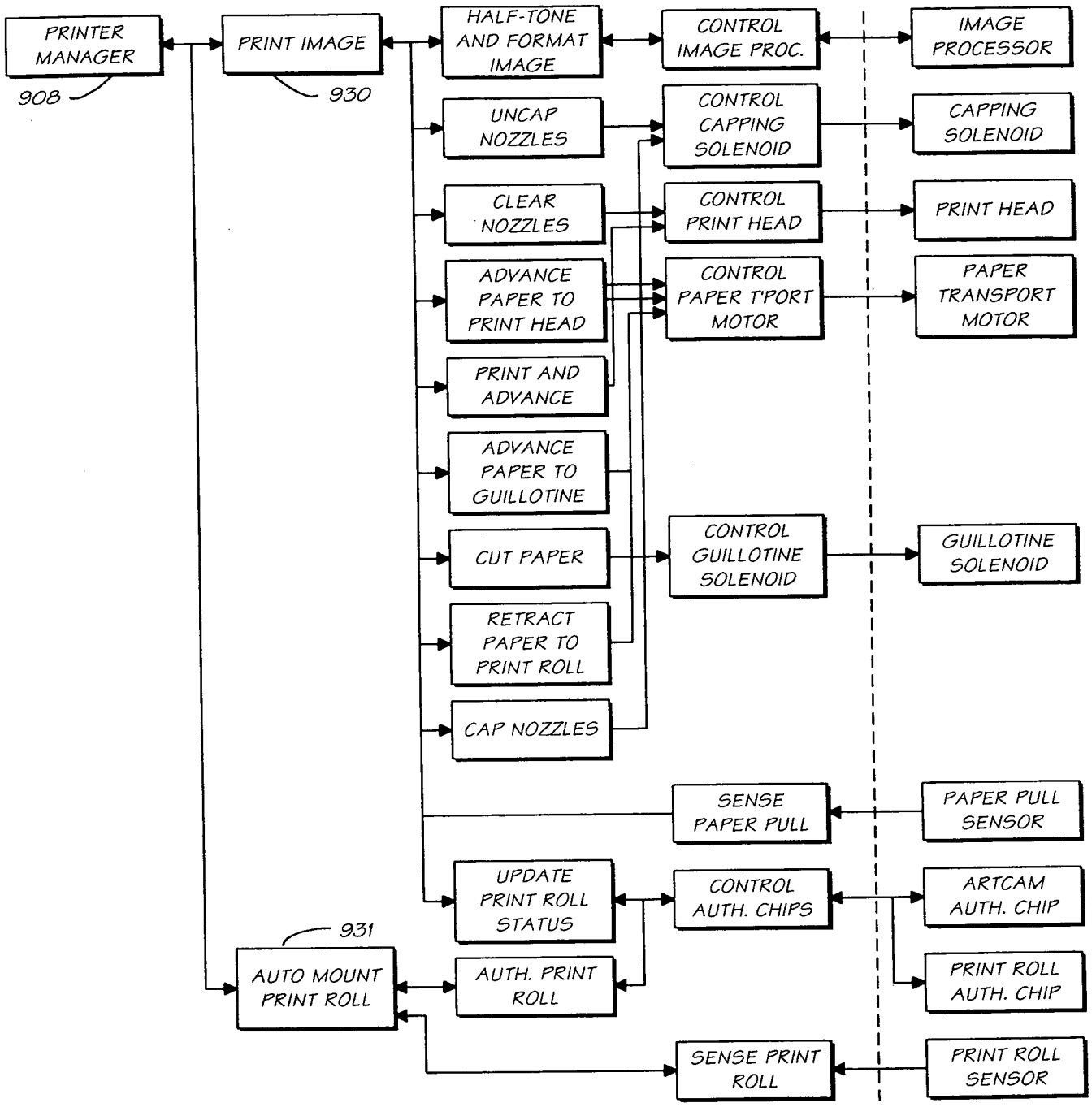

PRINTER MANAGER
908
PRINT IMAGE
930
HALF-TONE AND FORMAT IMAGE
CONTROL IMAGE PROC.
IMAGE PROCESSOR
UNCAP NOZZLES
CONTROL CAPPING SOLENOID
CAPPING SOLENOID
CLEAR NOZZLES
CONTROL PRINT HEAD
PRINT HEAD
ADVANCE PAPER TO PRINT HEAD
CONTROL PAPER T'PORT MOTOR
PAPER TRANSPORT MOTOR
PRINT AND ADVANCE
ADVANCE PAPER TO GUILLOTINE
CUT PAPER
CONTROL GUILLOTINE SOLENOID
GUILLOTINE SOLENOID
RETRACT PAPER TO PRINT ROLL
CAP NOZZLES
SENSE PAPER PULL
PAPER PULL SENSOR
UPDATE PRINT ROLL STATUS
CONTROL AUTH. CHIPS
ARTCAM AUTH. CHIP
931
AUTO MOUNT PRINT ROLL
AUTH. PRINT ROLL
PRINT ROLL AUTH. CHIP
SENSE PRINT ROLL
PRINT ROLL SENSOR

FIG. 224

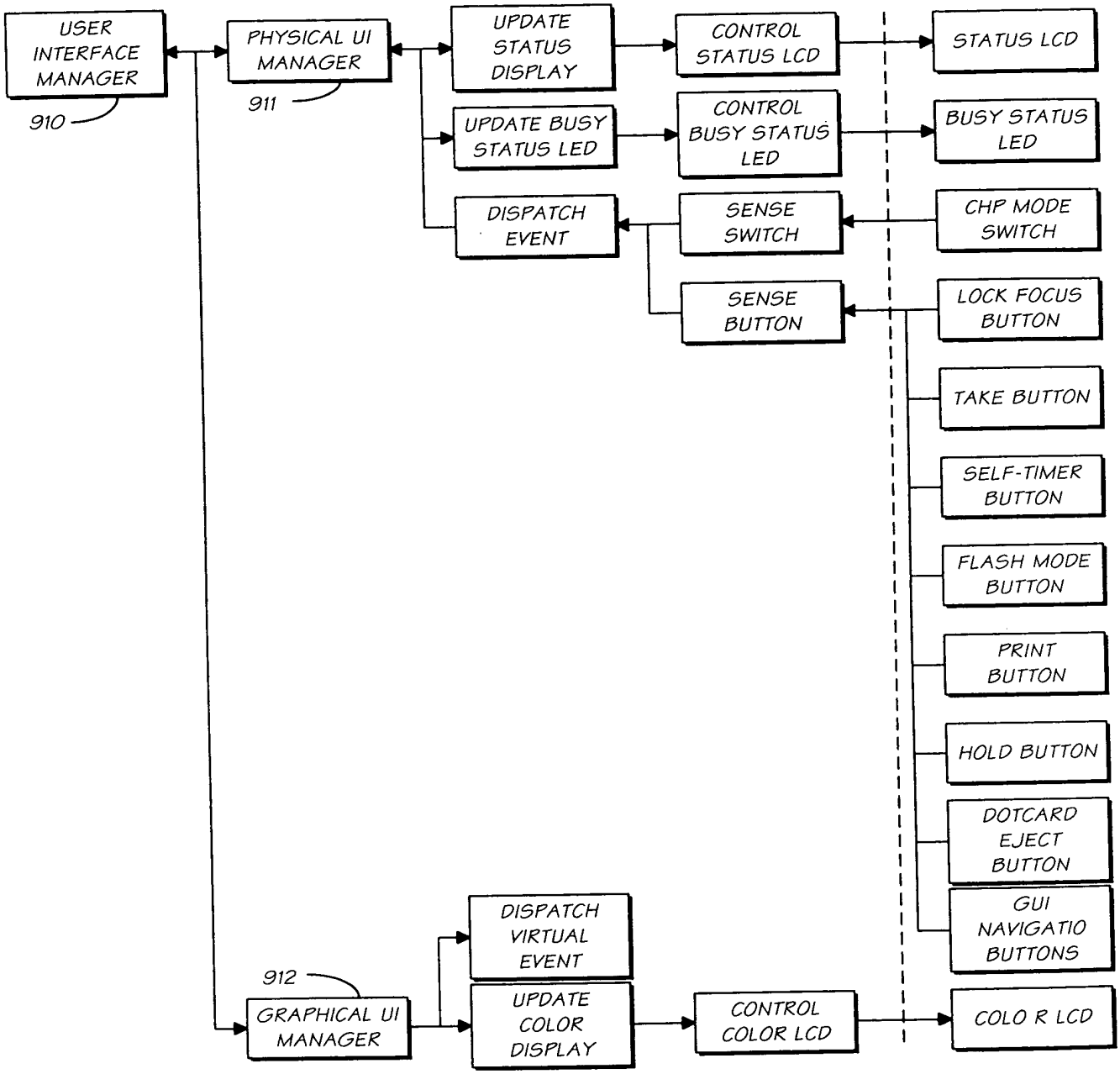

USER INTERFACE MANAGER
910
PHYSICAL UI MANAGER
911
UPDATE STATUS DISPLAY
UPDATE BUSY STATUS LED
DISPATCH EVENT
CONTROL STATUS LCD
CONTROL BUSY STATUS LED
SENSE SWITCH
SENSE BUTTON
STATUS LCD
BUSY STATUS LED
CHP MODE SWITCH
LOCK FOCUS BUTTON
TAKE BUTTON
SELF-TIMER BUTTON
FLASH MODE BUTTON
PRINT BUTTON
HOLD BUTTON
DOTCARD EJECT BUTTON
GUI NAVIGATIO BUTTONS
912
GRAPHICAL UI MANAGER
DISPATCH VIRTUAL EVENT
UPDATE COLOR DISPLAY
CONTROL COLOR LCD
COLO R LCD

FIG. 225

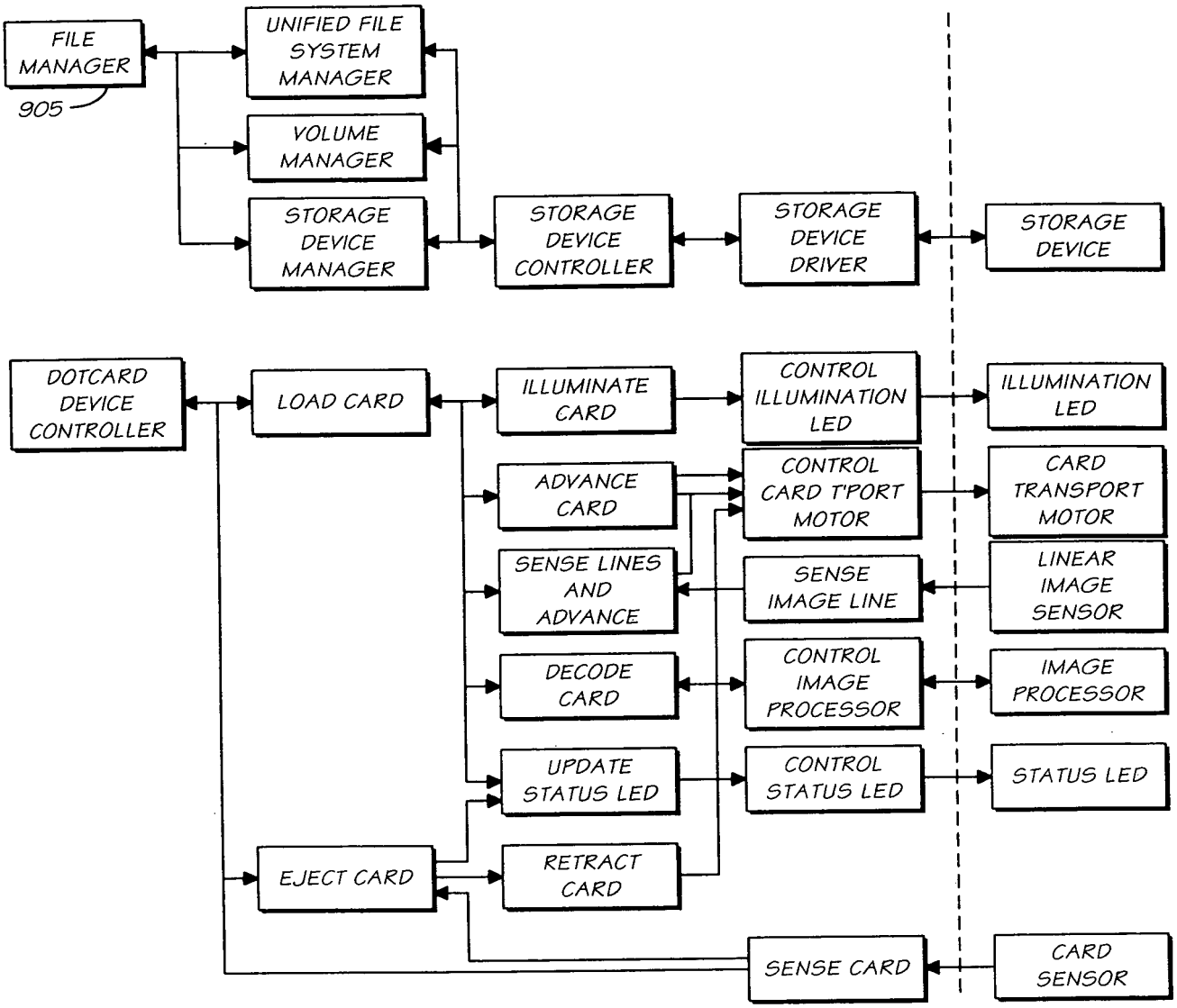

FILE MANAGER
905
UNIFIED FILE SYSTEM MANAGER
VOLUME MANAGER
STORAGE DEVICE MANAGER
STORAGE DEVICE CONTROLLER
STORAGE DEVICE DRIVER
STORAGE DEVICE
DOTCARD DEVICE CONTROLLER
LOAD CARD
ILLUMINATE CARD
ADVANCE CARD
SENSE LINES AND ADVANCE
DECODE CARD
UPDATE STATUS LED
EJECT CARD
RETRACT CARD
CONTROL ILLUMINATION LED
CONTROL CARD T'PORT MOTOR
SENSE IMAGE LINE
CONTROL IMAGE PROCESSOR
CONTROL STATUS LED
SENSE CARD
ILLUMINATION LED
CARD TRANSPORT MOTOR
LINEAR IMAGE SENSOR
IMAGE PROCESSOR
STATUS LED
CARD SENSOR

FIG. 226

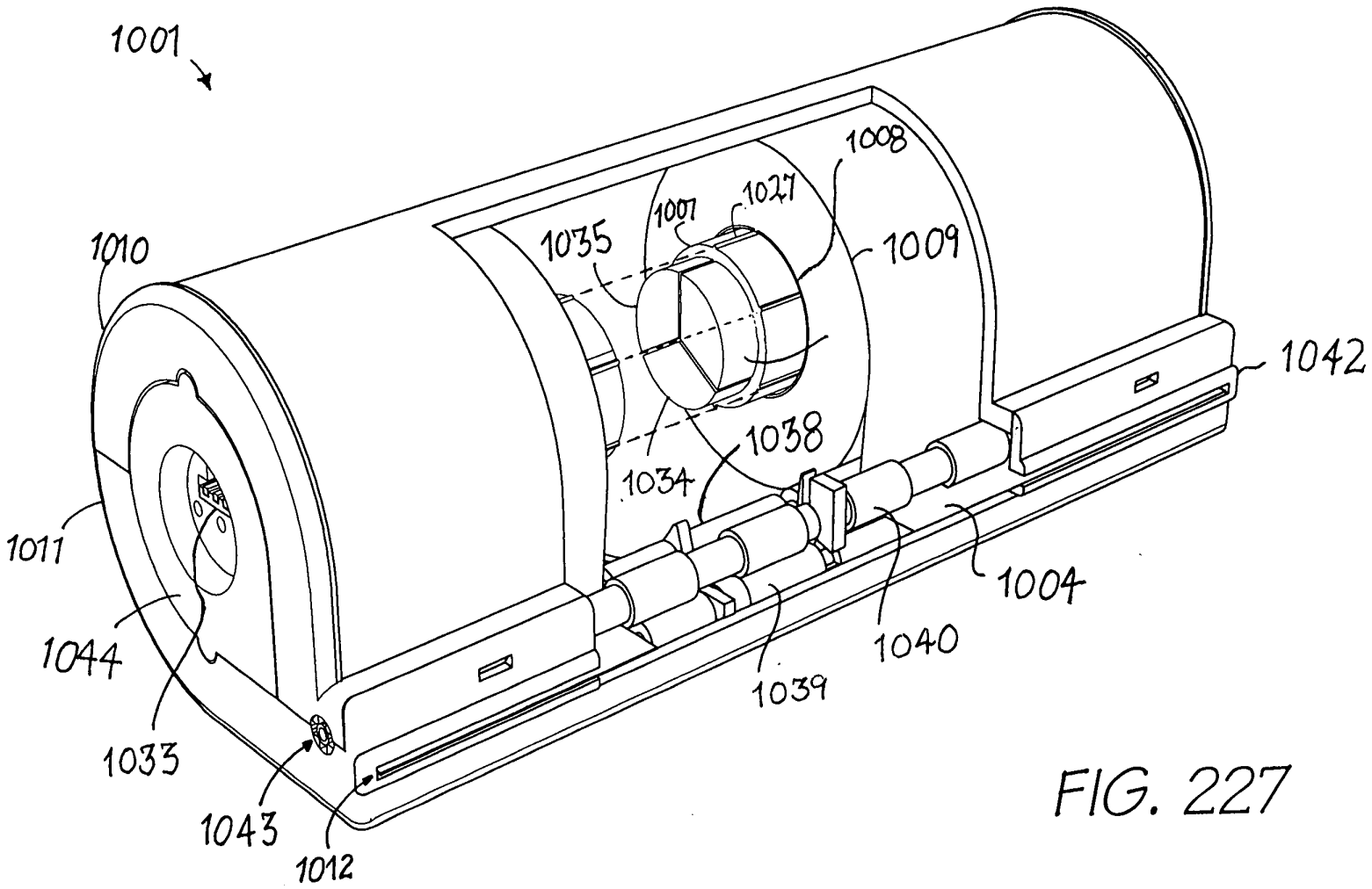

1001
1010
1035
1007
1027
1008
1009
1042
1038
1034
1011
1044
1033
1043
1012
1039
1040
1004

FIG. 227

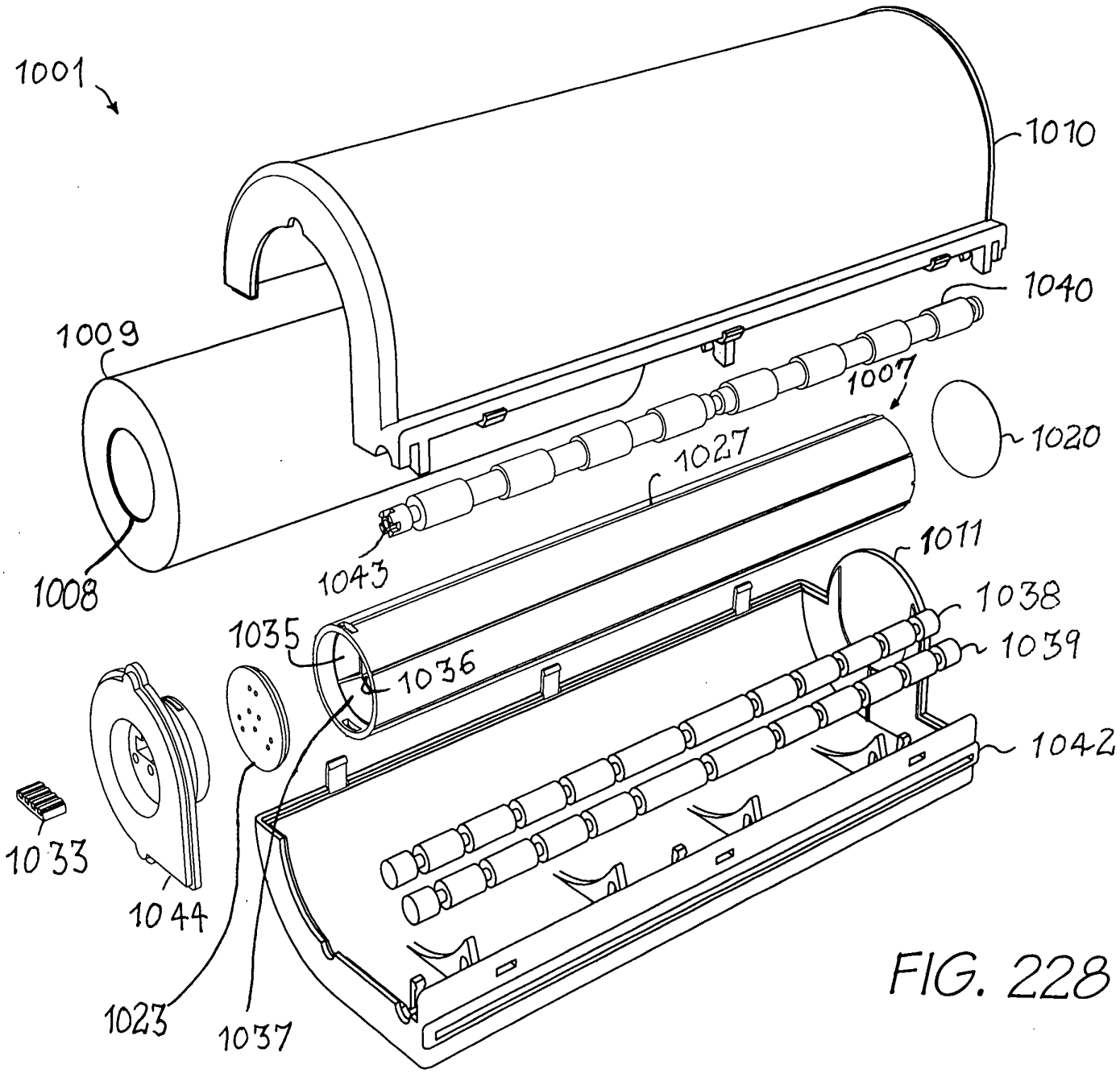

1001
1010
1040
1009
1007
1020
1027
1011
1008
1043
1038
1035
1039
1036
1042
1033
1044
1023
1037

FIG. 228

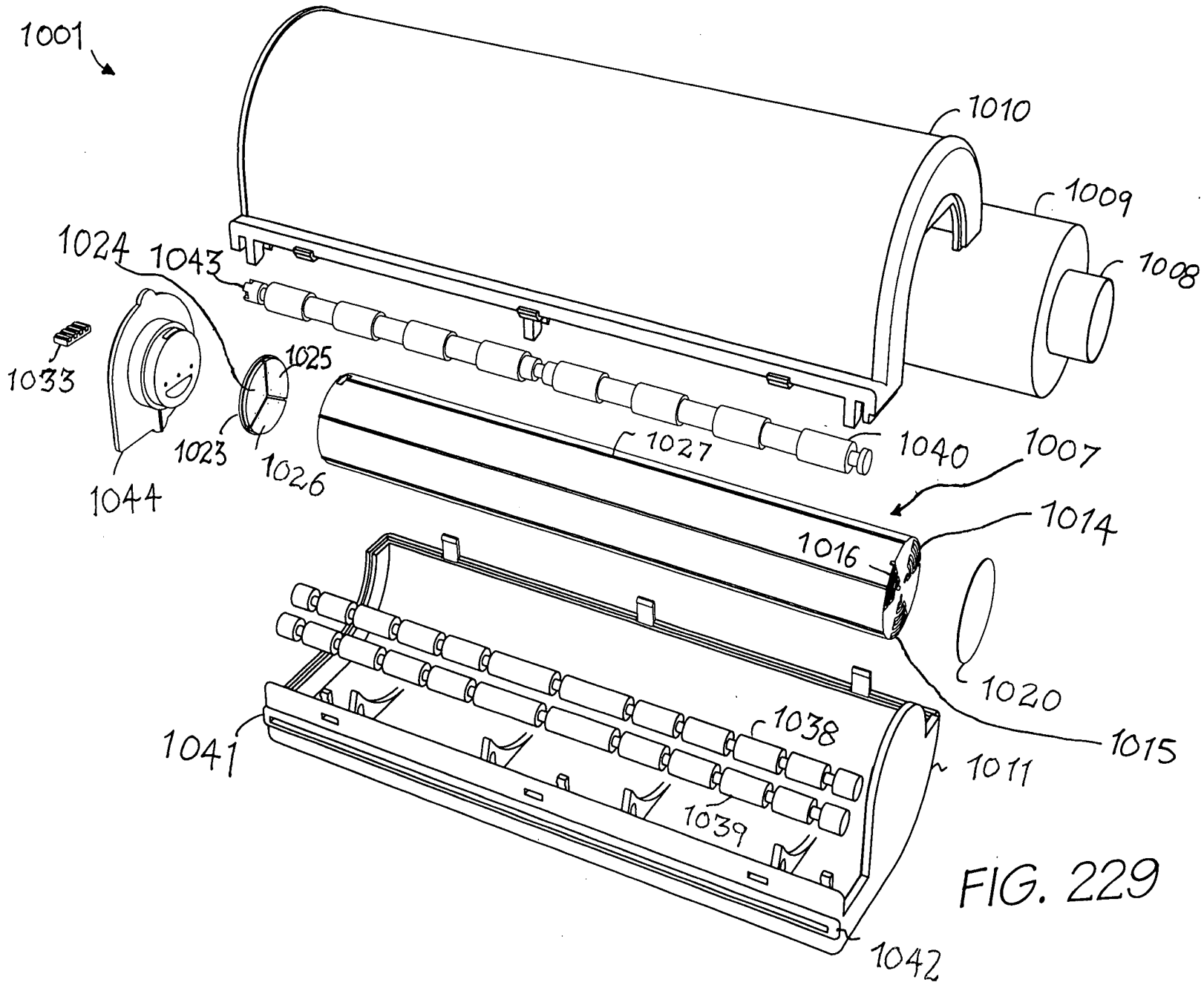

1001
1010
1009
1008
1024
1043
1033
1025
1023
1026
1044
1027
1040
1007
1014
1016
1020
1015
1038
1041
1011
1039
1042

FIG. 229

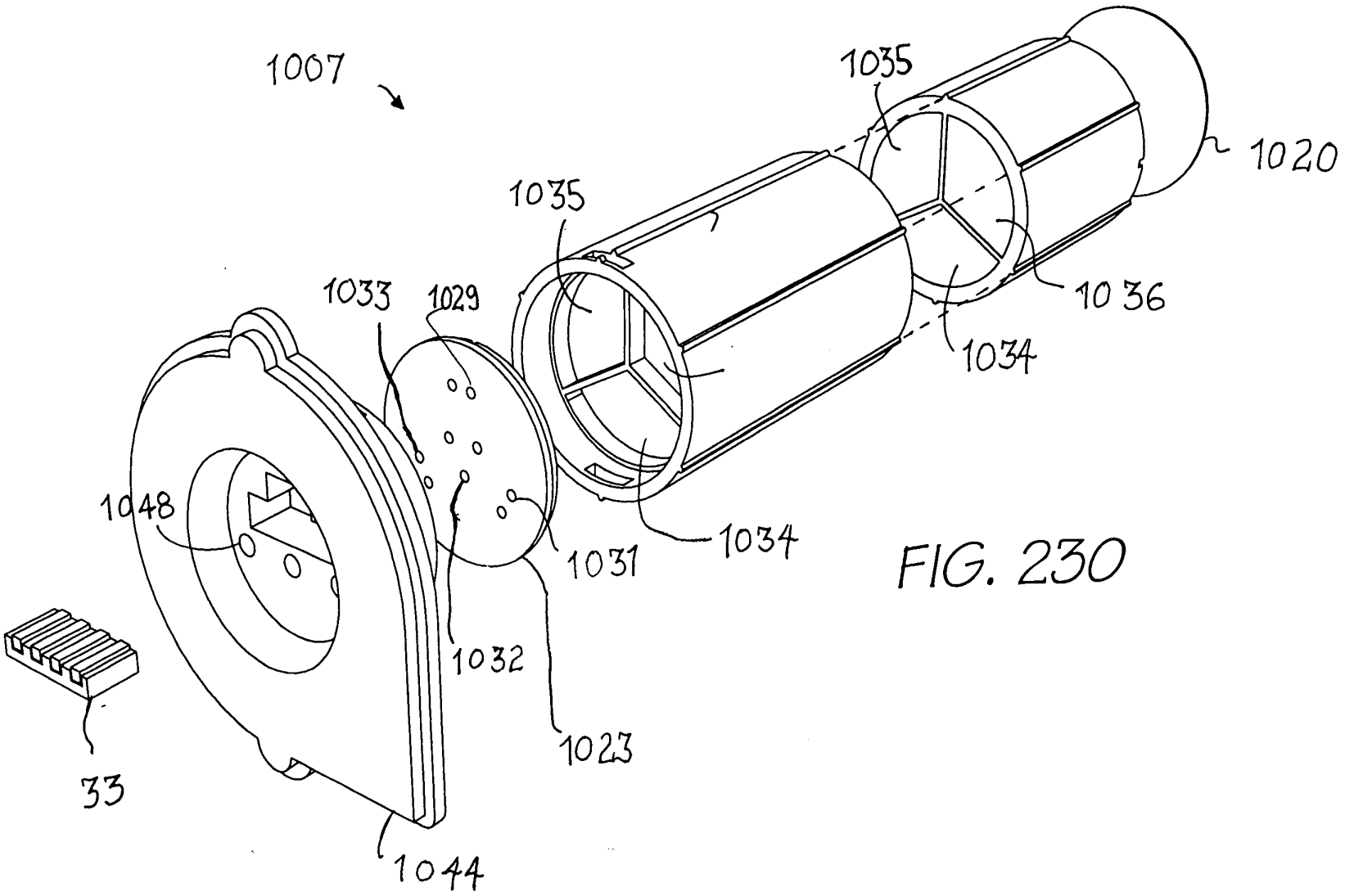

1007
1035
1020
1035
1033
1029
1036
1034
1048
1034
1031
1032
1023
33
1044

FIG. 230

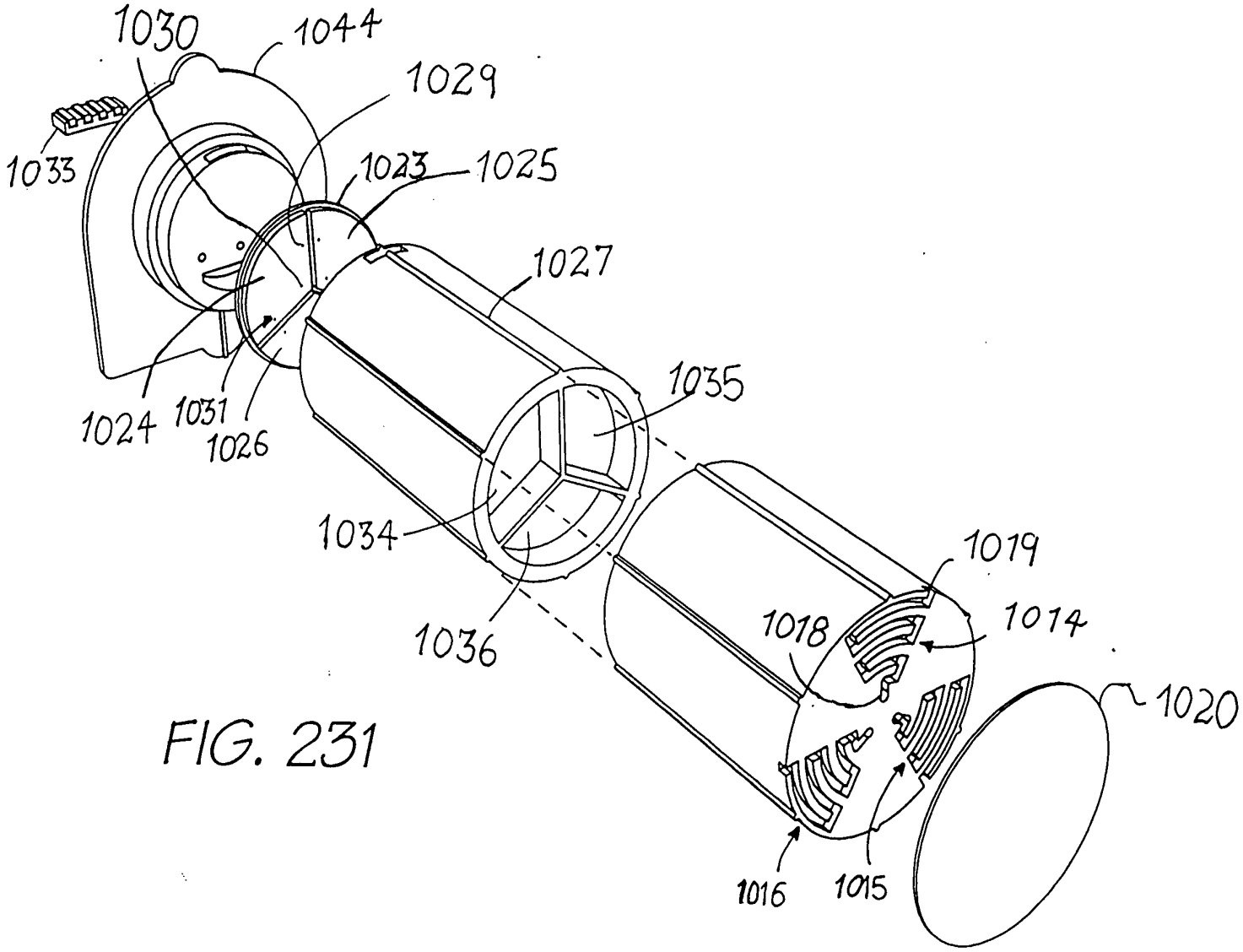

1030
1044
1029
1033
1023
1025
1027
1035
1024
1031
1026
1034
1019
1018
1014
1036
1020
1016
1015

FIG. 231